

Choose me!: Ampliación de un sistema de gestión de la elección de docencia

Mario Flores Romo

GRADO EN INGENIERÍA INFORMÁTICA. FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID



TRABAJO DE FIN DE GRADO

CURSO 2019-2020

Directores:

Antonio Sarasa Cabezuelo

Pablo Rabanal Basalo

Choose me!: Expansion of a teaching choice system

Mario Flores Romo

COMPUTER ENGINEERING DEGREE. FACULTY OF COMPUTER SCIENCE
COMPLUTENSE UNIVERSITY OF MADRID



FINAL DEGREE PROJECT

COURSE 2019-2020

Directors:

Antonio Sarasa Cabezuelo

Pablo Rabanal Basalo

Resumen

Este trabajo de fin de grado trata de una ampliación de un proyecto que empezó el curso anterior cuyo objetivo era el de automatizar las tareas que conllevan la elección de docencia de los profesores del departamento de Sistemas Informáticos y Computación de la Facultad de Informática de la Universidad Complutense de Madrid.

Actualmente, la elección de docencia se lleva a cabo de manera manual mediante hojas de cálculo. Debido a la complejidad y el esfuerzo que necesita esta tarea, para este trabajo fin de grado se ha continuado el desarrollo del sistema añadiendo la posibilidad de almacenar la docencia de los profesores año tras año, la gestión de las reservas y de las muestras de interés hacia ciertas asignaturas. De esta forma, los profesores podrán asegurar impartir algunas asignaturas para el próximo año académico y no depender de lo que hayan escogido los compañeros de docencia con mayor escalafón.

Se ha desarrollado una aplicación con la capacidad de poder utilizar el sistema en el futuro con otro tipo de clientes, implementando un servicio REST.

Palabras clave

Gestión de docencia, elección de asignaturas, aplicación web, planificación de horarios, API REST, Angular

Índice general

Índice	I
Índice de figuras	III
Agradecimientos	IV
Dedicatoria	V
1. Introducción	1
1.1. Motivaciones	1
1.2. Objetivo	2
1.3. Plan de trabajo	2
1.4. Estructura de la memoria	3
2. Introduction	5
2.1. Motivation	5
2.2. Objectives	6
2.3. Workplan	6
2.4. Structure of the report	7
3. Estado del arte	8
3.1. Sistema de planificación de horarios	8
3.1.1. BEST (Bullet Education Scheduling and Timetabling)	8
3.1.2. AscHorarios	9
3.2. Sistemas de recursos empresariales educativos	10
3.2.1. Odoo ERP Educación	10
3.2.2. Fedena	10
4. Tecnología empleada	12
4.1. Visual Studio Code	12
4.2. NPM	12
4.3. Angular	12
4.4. HTML5 y CSS3	13
4.5. Apache	13
4.6. SQLite	14
4.7. Django	14
4.8. virtualenv	14
4.9. openpyxl	14

5. Casos de uso	15
5.1. Actores de la aplicación	15
5.2. Casos de uso	16
5.2.1. Módulo histórico	16
5.2.2. Módulo gestión de reservas	18
5.2.3. Módulo gestión de muestras de interés	21
6. Modelo de datos	25
6.1. Modelo Entidad-Relación	25
6.2. Implementación	27
6.2.1. Histórico	27
6.2.2. Gestión de reservas	28
6.2.3. Gestión de muestras de interés	29
7. Arquitectura de la aplicación	31
7.1. Patrones de diseño	32
7.1.1. Capa de presentación	32
7.1.2. Servicio REST	32
8. Diseño de la aplicación	34
8.1. Funcionalidad de la aplicación	34
8.1.1. Histórico	34
8.1.1.1. Exportar histórico	36
8.1.1.2. Importar histórico	37
8.1.2. Gestión de reservas	37
8.1.2.1. Vista administrador	41
8.1.3. Gestión de muestras de interés	44
8.1.3.1. Vista administrador	45
8.1.4. Elección de docencia	46
9. Conclusiones y trabajo futuro	48
9.1. Conclusiones	48
9.2. Trabajo futuro	48
10. Conclusions and future work	50
10.1. Conclusions	50
10.2. Future work	50
A. Guía de uso	54
A.1. Consultar histórico	54
A.2. Exportar histórico	55
A.3. Importar histórico	56
A.4. Añadir reserva	57
A.5. Eliminar reserva	59

A.6. Ver reservas	59
A.7. Administrador - Aceptar reserva	60
A.8. Administrador - Rechazar reserva	63
A.9. Administrador - Deshacer reserva	64
A.10. Añadir muestra de interés	65
A.11. Eliminar muestra de interés	67
A.12. Ver muestras de interés	68
A.13. Administrador - Aceptar muestra de interés	69
A.14. Administrador - Rechazar muestra de interés	70
A.15. Administrador - Deshacer muestra de interés	71
B. Guía de instalación	73
B.1. Instalación de la aplicación web	73
B.2. Instalación de la API	74

Agradecimientos

Lo primero, agradecer a los tutores del proyecto, Antonio Sarasa Cabezuelo y Pablo Rabanal Basalo, sin ellos este trabajo hubiera sido un despropósito. Gracias a sus pautas, consejos y directrices este proyecto ha conseguido salir adelante.

También quisiera agradecer el apoyo y la calma prestadas por mi padre, muchas veces me he visto en un túnel sin luz al final del camino y él me ha dado la capacidad de abrir los ojos para verla. A mis amigos por la insistencia mostrada en saber la evolución del trabajo, deseosos de verme al fin con mi grado universitario.

Dedicatoria

A Belén y Miguel.

A Jorge.

Capítulo 1

Introducción

En esta introducción se hablará de los motivos del desarrollo de la aplicación, de cuáles son los objetivos del proyecto y del plan de trabajo seguido.

1.1. Motivaciones

En el departamento de Sistemas Informáticos de nuestra facultad, la elección de la docencia de los profesores se lleva a cabo mediante un sistema rudimentario de hojas de cálculo en el que cada uno de los miembros se encarga de realizar sus elecciones de horarios y comunicárselas a un coordinador común. Dicho coordinador se encarga posteriormente de comprobar compatibilidades de horarios, prioridades de los profesores según antigüedad y otros muchos factores que convierten este proceso en un trabajo altamente tedioso y complejo, que lleva aproximadamente 3 días.

Observada la obsolescencia de dicho método, se propone el desarrollo de una aplicación web que agilice el proceso tanto para profesores como para el coordinador, facilitando la comunicación entre ambas partes y automatizando ciertas funciones y comprobaciones que previamente tenían que ser realizadas manualmente.

En el año 2018, dos compañeros de la facultad comenzaron este proyecto [1] y desarrollaron una primera parte en la que se asentaban las bases de la estructura del proyecto, las tecnologías a utilizar, etc. Además, desarrollaron los primeros y fundamentales módulos de la aplicación. Éstos fueron los módulos funcionales de las asignaturas y los profesores,

pudiendo registrar nuevos objetos, editar los ya existentes, o eliminarlos. También se encargaron de la autenticación de la aplicación y comenzaron el desarrollo del módulo más importante dentro de la aplicación, la elección de docencia.

1.2. Objetivo

El objetivo de este proyecto es la continuación del desarrollo de la aplicación web. Los objetivos concretos son:

- Habilitar y gestionar la posibilidad de realizar reservas de la docencia de asignaturas, de manera que se facilite la labor del administrador y se agilice el proceso para el profesor, evitando estar a expensas de que el administrador esté disponible para una hipotética reunión.
- Habilitar y gestionar la posibilidad de mostrar interés hacia una asignatura. Existen ciertas asignaturas que su docencia se elige por una vía distinta a la ordinaria y esta funcionalidad permitirá al administrador automatizar esta tarea.
- Implementar un histórico de asignaturas y profesores. Dado que algunas de las acciones realizables en la aplicación tienen dependencia de elecciones tomadas otros años, es imprescindible el mantenimiento de un histórico que permita y facilite al administrador el acceso y consulta de la docencia impartida en el pasado, dado que dicha información es relevante para permitir o denegar peticiones de reservas.

1.3. Plan de trabajo

Para llevar a cabo el proyecto, se ha considerado importante planificar el desarrollo del mismo. Esta planificación es motivada por asegurar el control del trabajo, así como tener una referencia preconcebida del proyecto con la que poder contrastar el avance de la aplicación.

Se ha planteado un orden específico de implementación de los objetivos para el desarrollo del proyecto. El orden es el siguiente:

1. Desarrollo del módulo histórico.
2. Desarrollo del módulo reserva (solo funcionalidades de los profesores).
3. Desarrollo del módulo reserva (funcionalidades de administrador).
4. Implementación del módulo muestra de interés.
5. Integrar los nuevos módulos a la elección de docencia.

Para cada una de los puntos, se ha seguido el mismo patrón de trabajo. Primero se analizan los requisitos de la funcionalidad, produciendo los diagramas de casos de uso. Tras ello, se definen las tablas necesarias y se añaden a la base de datos. Posteriormente, se pasará a la implementación. Finalmente, se consultará a los tutores del proyecto sobre su funcionamiento y estética.

1.4. Estructura de la memoria

El presente trabajo se estructura en los siguientes capítulos. Los dos primeros capítulos son la introducción del trabajo, el primero de ellos escrito en castellano y el segundo en inglés. En el capítulo 3, se exponen los productos similares a este trabajo que existen en el mercado. Después, el capítulo 4 describe las tecnologías empleadas para el desarrollo del proyecto. El capítulo 5 describe los casos de uso de las funcionalidades desarrolladas de la aplicación, presentados en 3 módulos: histórico, gestión de las reservas y gestión de las muestras de interés. Más tarde, en el sexto capítulo se explica la estructura de la base de datos de la aplicación, donde se encuentra el modelo entidad-relación, así como una explicación detallada de las tablas utilizadas. El capítulo 7 habla de los patrones de diseño y la arquitectura utilizada para el desarrollo del proyecto. En el octavo capítulo, se detalla el diseño y la funcionalidad de la aplicación, aportando capturas de la web y fragmentos de código. Las conclusiones del proyecto y el trabajo futuro se expondrán en el noveno capítulo en castellano y en el décimo en inglés. En el último lugar, se encuentran los apéndices A y

B. En el primero se encuentra la guía de uso de la aplicación y en el segundo se incluyen las guías de instalación de la web y de la API.

Capítulo 2

Introduction

This introduction will discuss the reasons for the implementation and the objectives of the project and the work plan followed.

2.1. Motivation

In the Department of Computer Systems and Computing of our faculty, the teaching assignment is made through a rudimentary system of spreadsheets in which each member is responsible for making his/her own timetable choices and communicating them to a common coordinator. This coordinator is then responsible for checking the compatibility of timetables, teachers' priorities according to seniority and many other factors that make this process a highly tedious and complex task which takes approximately 3 days.

Noting the obsolescence of this method, it is proposed to develop a web application that will speed up the process for both teachers and the coordinator, facilitating communication between the two parties and automating certain functions and checks that had previously to be performed manually.

In 2019, two colleagues of the faculty began this project [1] and they developed a first part that laid the foundations for the structure of the project, the technologies to be used, etc. In addition, they developed the first and fundamental modules of the application. These were the functional modules of the subjects and teachers, being able to register new objects, edit those already existing, or eliminate them. They also completed the authentication of

the application, and they began the development of the most important module within the application, the teaching assignment.

2.2. Objectives

The objective of this project is to continue the development of the web application. The specific objectives are:

- Enable and manage the possibility of making teaching reservations for subjects, so as to facilitate the work of the administrator and speed up the process for the teacher, avoiding being at the expense of the administrator being available for a hypothetical meeting.
- Enable and manage the ability to show interest in a subject. There are certain subjects that their teaching is chosen in a different way than the ordinary one, and this functionality will allow the administrator to automate this task.
- Implement a history of subjects and teachers. Since some of the implementable actions depend on choices made in other years, it is essential to maintain a history that allows and facilitates the administrator access and consultation of the teaching given in recent years, since such information is highly relevant for allowing or denying requests for reservations.

2.3. Workplan

In order to carry out the project, it has been considered important to plan the development of the project. This planning is motivated by ensuring the control of the work, as well as having a preconceived reference of the project with which to compare the progress of the application.

A specific order of implementation of the objectives for the development of the project has been proposed. The order is as follows:

1. Development of the historical module.
2. Development of the reserve module (only functionalities of teachers).
3. Development of the reserve module (administrator functionalities).
4. Module implementation shows interest.
5. Integrate new modules into teaching choices.

For each of the points, the same pattern of work has been followed. First the functionality requirements are analyzed, producing the use case diagrams. After that, the necessary tables are defined and added to the database. Subsequently, the implementation will be moved. Finally, the project tutors will be consulted on its operation and aesthetics.

2.4. Structure of the report

The present work is structured in the following chapters. The first two chapters are the introduction to the work, the first of them written in Spanish and the second in English. In Chapter 3, the products similar to this work that exist in the market are exposed. Later, Chapter 4 describes the technologies used for the development of the project. Chapter 5 describes the cases of use of the functionalities developed from the application, presented in 3 modules: history, management of reservations and management of samples of interest. The structure of the implementation database is explained later in the sixth chapter where it is found the entity-relationship model, as a detailed explanation of the tables used. Chapter 7 discusses the design patterns and architecture used for project development. The eighth chapter details the design and functionality of the application, providing web captures and code snippets. The conclusions of the project and future work will be presented in the ninth chapter in Spanish and the tenth in English. In the last place, you will find Appendices A and B. In the first you will find the application usage guide, and in the second you will find the web installation guides and the API.

Capítulo 3

Estado del arte

Tras esta breve introducción, se puede destacar el hecho de que el proyecto está destinado a ser un gestor de planificación docente para un conjunto de profesores de una facultad. En este sector, el de sistemas de gestión, encontramos dos grandes grupos de software que son capaces de resolver el problema presentado: el software dirigido a la planificación de horarios y los sistemas de planificación de recursos empresariales (ERP).

3.1. Sistema de planificación de horarios

Este tipo de sistemas tienen como cualidades el poder de especificar cualquier restricción, combinar y evaluar varios objetivos de acuerdo a los intereses de la institución, optimizando los horarios para el profesorado, los estudiantes y las salas/aulas.

Dentro de este grupo de sistemas, destacan algunos software que tienen un hueco en el mercado. Estos son:

3.1.1. BEST (Bullet Education Scheduling and Timetabling)

Es un software de planificación de horarios enfocado sobre todo a universidades. Además, es capaz de gestionar en tiempo real los eventos que vayan surgiendo conforme el año escolar avanza y por consiguiente, los recursos necesarios para poder realizarlos [2]. En estos momentos, se encuentran desarrollando una nueva funcionalidad que les permita gestionar el cuidado de exámenes. Sus propiedades principales son:

- Gestionar horarios de profesores, alumnos y aulas.
- Planificación y gestión de eventos y recursos relacionados directamente con la universidad.
- Panel de indicadores de rendimiento, permitiendo así una rápida visión de la información.
- Dota de un portal web donde todos los clientes de la aplicación (profesores y alumnos) pueden gestionar su docencia.
- Sistema de soporte de decisiones.

3.1.2. AscHorarios

Al igual que el anterior, es un software especializado en planificar horarios de centros educativos [3]. Cabe señalar que cuenta con una interfaz muy intuitiva que hace de su uso algo sencillo. También ofrece la posibilidad de introducir los horarios del centro de manera digital, haciendo de este proceso algo mucho más liviano. Cuenta con una aplicación de móvil, con lo que consigue que sea muy rápido de gestionar. Se caracteriza por:

- Generación automática de horarios, con la posibilidad de modificarlos manualmente.
- Es compatible con clases en distintos edificios y puede optimizar el traslado de uno a otro.
- Tiene una funcionalidad que se encarga de las sustituciones de los profesores.
- Capacidad para importar datos digitalizados.
- Se adapta a cualquier restricción.

3.2. Sistemas de recursos empresariales educativos

Los ERP educativos proporcionan a las instituciones educativas la eficacia que necesitan permitiendo la gestión automatizada de las responsabilidades administrativas y la gestión tanto de docentes como estudiantes. Se han desarrollado tanto estos sistemas, que a día de hoy son capaces de ofrecer infinidad de funcionalidades eficaces, como son el control de pagos por parte del alumnado, gestión de las matrículas, administración de reportes, etc. Existen muchos ERP en el mercado, pero hemos destacado estos dos:

3.2.1. Odoo ERP Educación

Es un ERP que muchas personas utilizan en su día a día para su empresa, pero que se puede llevar al ámbito educativo [4]. En éste, ofrecen multitud de funcionalidades que facilitan la administración digital de un centro educativo como puede ser una universidad. Añadir que es un sistema publicado bajo licencia *open source*. Sus funciones principales son:

- Gestión de estudiantes: información personal, gestión de la matrícula, asistencia a clase, etc.
- Organización de calendario y tablas de horarios.
- Gestión de la biblioteca: préstamo de libros, reserva de salas, etc.
- Gestión de exámenes.
- Posibilidad de trabajar en la nube.

3.2.2. Fedena

Es un sistema de planificación orientado al ámbito escolar. Al igual que el ejemplo anterior, Fedena es un proyecto *open source* donde todo el que quiera, puede aportar nuevas ideas [5]. Se centra fundamentalmente en la gestión de registros y admisiones para los centros educativos. La aplicación ofrece módulos de: calendario, gestión financiera, gestión de exámenes e inicios de sesión de estudiantes y padres, etc. Sus principales características son:

- Calendario de eventos.
- Reporte a los padres vía SMS.
- Gestión de horarios.
- Gestión del cuidado de los exámenes.

Capítulo 4

Tecnología empleada

En este capítulo se numeran las herramientas utilizadas para desarrollar la aplicación. Todas las tecnologías empleadas en este proyecto fueron elegidas por los compañeros que hicieron la primera parte.

4.1. Visual Studio Code

El proyecto se ha desarrollado con el editor Visual Studio Code [6]. Las razones por las que fue elegido son su control de versiones y la paleta de colores que se utiliza en el código. Además, ofrece una gran cantidad de extensiones con las que poder agilizar más el desarrollo.

4.2. NPM

Node Packet Manager [7], o comúnmente conocido como NPM, es el gestor de paquetes de Node.js por excelencia. Trabaja desde línea de comandos, se encarga de administrar los módulos de la aplicación y agregar dependencias de manera sencilla.

4.3. Angular

Angular es un *framework open source* mantenido por Google [8]. Es puntero en cuanto a desarrollo de webs SPA (del inglés *Singular Page Application*) se refiere. Separa de manera

evidente el *frontend* y el *backend*, consiguiendo así mayor control sobre la aplicación. Este control sostenido por el patrón MVC, que realmente en la práctica, se trata de un patrón MVVM [9], modelo-vista vista-modelo. La principal ventaja de este *framework*, a parte de las ya nombradas, es su modulación, proporcionando así gran facilidad a la hora de desarrollar el proyecto. Permite una navegación dinámica y fluida que consigue que no se recargue el navegador.

4.4. HTML5 y CSS3

En la capa de presentación hemos utilizado los lenguajes HTML5, CCS3 junto con el metalenguaje Sass.

HTML5 [10] es el lenguaje de marcas utilizado para desarrollar la aplicación. Esta versión, a parte de ser la más actual, añade nuevas etiquetas semánticas y proporciona una plataforma de desarrollo de complejas aplicaciones web. HTML [11] se encarga de dar nombre a los objetos que forman la aplicación web, ofreciendo así, un esbozo de lo que va a ser la página web. La forma se la aporta el segundo lenguaje mencionado al principio de este punto.

CSS3 [12] es el encargado de darle forma a la aplicación, aportando color, posición en la pantalla, etc. El lenguaje se rige por una serie de normas de etiquetado, utilizando las marcas del documento HTML para referirse a un elemento en concreto, o a una serie de elementos que compartan una característica.

Sass [13] es un lenguaje especializado que aporta a CSS funcionalidades que no sería capaz de conseguir si no fuera por él, como puede ser la herencia de reglas.

4.5. Apache

Apache es un servidor web gratuito y *open source* [14]. Apache dispone de módulos para dar soporte a múltiples lenguajes, como Perl, Python o PHP. Además, es muy popular por la gran cantidad de configuraciones y módulos que amplían sus funcionalidades.

4.6. SQLite

SQLite [15] es un motor de bases de datos relacional de transacciones de código abierto, de configuración muy sencilla, autónomo e independiente diseñado para integrarse en una aplicación. También permite que una única conexión de base de datos acceda a múltiples archivos de base de datos simultáneamente.

4.7. Django

Django [16] es un *framework* web *open source* completamente funcional y está escrito en Python. La elección de este *framework* viene dada porque es seguro, proporcionando una manera segura de administrar cuentas de usuario y contraseñas. Además ofrece protección ante algunas vulnerabilidades, como SQL Injection. Otro gran pilar es su mantenibilidad. Éste fomenta la reutilización de código, agregándose al principio "*Don't repeat yourself*". Como última característica que destacar del *framework* es la posibilidad de realizar una consulta a la base de datos sin utilizar SQL.

4.8. virtualenv

virtualenv [17] es una herramienta de desarrollo en Python usada para crear entornos aislados, lo que posibilita la instalación de distintos paquetes sin que haya ningún tipo de interferencia.

4.9. openpyxl

openpyxl [18] es una librería de Python que permite operar con archivos Excel. Lo que caracteriza a esta librería *open source* es que permite crear fórmulas, añadir, eliminar y modificar tablas.

Capítulo 5

Casos de uso

En este capítulo se van a describir los casos de uso definidos para la aplicación desarrollada agrupados en módulos funcionales: histórico, gestión de reservas y gestión de muestras de interés. También se describen a los actores principales de estos casos de uso.

5.1. Actores de la aplicación

Los actores definidos en los casos de uso que se muestran en este capítulo son:

- **Profesor:** Representa a un profesor del departamento. Su función principal es realizar su elección docencia. Tiene permisos de edición en todas las funcionalidades en lo que respecta a él mismo (su perfil, reservas, elección, etc.).
- **Administrador:** Representa a aquella persona que dispone del control total de la herramienta, de modo que tiene todos los permisos para editar y eliminar cualquier información, desde los perfiles de usuario hasta las elecciones docentes. Intervendrá en todos los procesos de implementados en la aplicación: elección de docencia, reservas y muestras de interés, así como cualquier evento excepcional que necesite solución.

5.2. Casos de uso

5.2.1. Módulo histórico

El módulo histórico tiene como función albergar la información sobre las elecciones docentes de los últimos 7 años. Además, supondrá la base para el desarrollo del módulo de gestión de reservas.

A continuación, se especifican cada uno de los casos de uso de este módulo (ver figura 3.1).

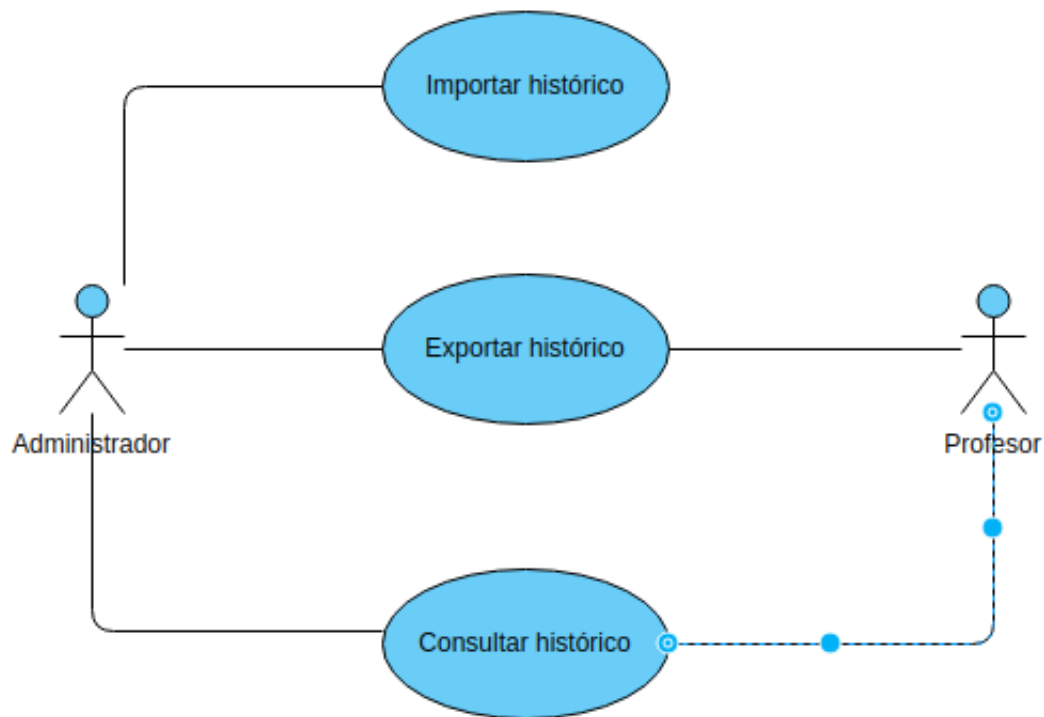


Figura 5.1: *Diagrama de casos de uso del módulo histórico*

DOCENCIA01	Consultar histórico
Actor	Profesor y administrador
Precondición	El usuario debe tener una cuenta ya registrada y tiene que haberse identificado.
Secuencia normal	1. Se ofrecen una serie de opciones para filtrar la búsqueda. 2. Se muestran las filas responden a la consulta.
Postcondición	Aparecen los resultados requeridos en la tabla.
Excepciones	Si es el administrador quien realiza esta acción, se añade la opción de búsqueda: profesor.

DOCENCIA02	Importar histórico
Actor	Administrador
Descripción	Importa de un documento Excel los horarios de los profesores con sus asignaturas, aulas, etc.
Precondición	Se necesita tener un documento Excel con el formato impuesto por la aplicación para su correcto funcionamiento. Los archivos de importación deben contener el histórico de un año completo.
Secuencia normal	1. El sistema muestra un campo para subir el archivo. 2. Subida del fichero al sistema. 3. Mensaje en la aplicación informando que la subida ha sido un éxito.
Postcondición	El histórico de ese año se llena con los datos del archivo en caso de no tener información y lo sobrescribe en caso contrario.

DOCENCIA03	Exportar histórico
Actor	Administrador
Descripción	Exporta en un documento Excel la información consultada en la aplicación.
Precondición	Se debe haber hecho una consulta previamente y que la información aparezca en pantalla.
Secuencia normal	1. Descarga del fichero al usuario local. 2. <i>Pop-up</i> en la aplicación informando que la descarga ha sido un éxito.
Postcondición	Se genera un documento Excel (.xlsx) con la lista de todas las entradas que tenga la tabla histórico.
Excepciones	No se ha realizado ninguna búsqueda en el histórico, el sistema informa del error.

5.2.2. Módulo gestión de reservas

Este módulo se encarga de gestionar las reservas de asignaturas de cara a la elección docente. Esta funcionalidad pretende introducir en la aplicación la posibilidad de que los profesores puedan reservar asignaturas para simplificar el proceso de la elección. Existen una serie de restricciones de las que depende el que un profesor pueda o no reservar una asignatura. Estas restricciones son las siguientes:

- No se puede reservar una asignatura que sea de primer curso ni desdoble
- El profesor no podrá haber impartido esa asignatura más de dos años en los últimos seis.
- El profesor debe haber impartido la asignatura el año anterior a la realización de la reserva.

En la figura 5.2 se pueden observar los casos de uso de este módulo.

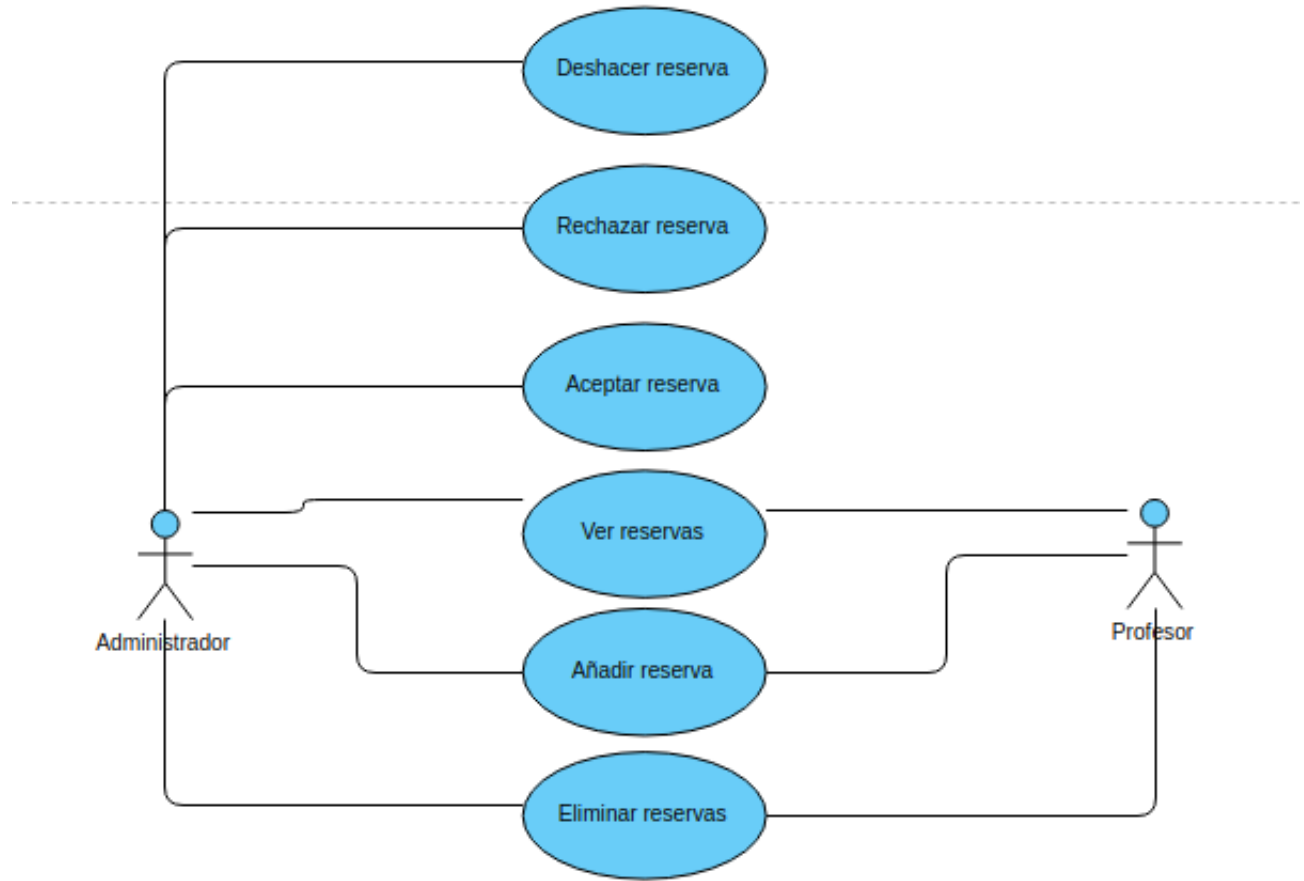


Figura 5.2: *Diagrama de casos de uso del módulo reserva docencia*

DOCENCIA04	Añadir reserva
Actor	Administrador y profesor
Descripción	Selecciona y guarda la reserva de una asignatura.
Precondición	El usuario debe tener una cuenta ya registrada y tiene que haberse identificado.
Secuencia normal	1. El sistema presenta las asignaturas reservables. 2. El actor escoge la asignatura que quiere reservar. 3. Finalmente, el actor puede añadir un comentario a la reserva.
Postcondición	El sistema guarda la reserva hecha por el profesor y queda pendiente de lo que decida el administrador.

DOCENCIA05	Eliminar reserva
Actor	Administrador y profesor
Descripción	Selecciona y borra la reserva de una asignatura.
Precondición	El profesor debe tener alguna reserva hecha para poder eliminarla.
Secuencia normal	1. El sistema presenta las asignaturas reservadas. 2. El profesor escoge la reserva que quiere eliminar.
Postcondición	El sistema elimina la reserva hecha por el profesor.
Excepciones	En caso de que la reserva esté tramitada, es decir, que el administrador haya decidido si la acepta o la rechaza, no podrá ser eliminada.

DOCENCIA06	Ver reservas hechas
Actor	Administrador y profesor
Descripción	El sistema presenta las reservas hechas por el profesor que ha iniciado sesión.
Precondición	El profesor debe tener una cuenta ya registrada y tiene que haberse identificado. Además debe haber realizado alguna reserva
Secuencia normal	1. El sistema presenta las asignaturas reservadas.
Postcondición	Devuelve las reservas del actor.

DOCENCIA07	Aceptar reserva
Actor	Administrador
Descripción	Acepta una reserva hecha por un profesor.
Precondición	Deben de haber realizado alguna reserva los profesores.
Secuencia normal	1. El sistema presenta las asignaturas reservadas por un profesor. 2. El administrador escoge la reserva que quiere aceptar. 3. En caso de que el número de reservas sea igual o mayor al número de grupos de la asignatura, el administrador le asigna un grupo. En caso contrario, será el profesor en su elección cuando lo decida.
Postcondición	El sistema guarda la reserva confirmada y se asigna la reserva a su elección docente.
Excepciones	En caso de que no queden grupos de la asignatura, directamente marcará la reserva como rechazada.

DOCENCIA08	Rechazar reserva
Actor	Administrador
Descripción	Rechaza una reserva hecha por un profesor.
Precondición	Deben de haber realizado alguna reserva los profesores.
Secuencia normal	1. El sistema presenta las asignaturas reservadas. 2. El administrador escoge la reserva que quiere rechazar.
Postcondición	El sistema elimina la reserva realizada por el profesor.

DOCENCIA09	Deshacer reserva
Actor	Administrador
Descripción	Deshace la decisión tomada sobre una reserva hecha por un profesor.
Precondición	Deben de haberse decidido alguna reserva.
Secuencia normal	1. El sistema presenta las asignaturas reservadas. 2. El administrador escoge la reserva decidida que quiere deshacer.
Postcondición	El sistema deshace la decisión impuesta sobre una reserva.

5.2.3. Módulo gestión de muestras de interés

Existen algunas asignaturas que son especiales, o de interés. Se conciben así debido a su procedimiento de elección. Las asignaturas que pertenecen a este grupo son: asignaturas de máster, grupos de inglés y asignaturas optativas. En la figura 5.3 se presentan los casos de uso del módulo.

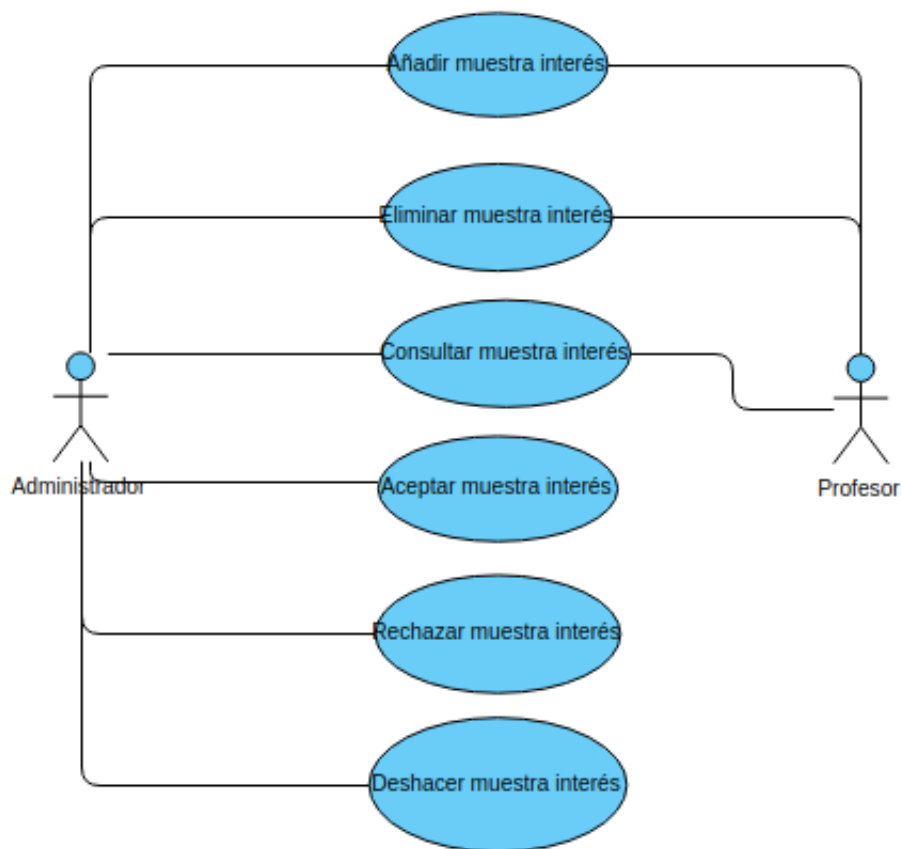


Figura 5.3: *Diagrama de casos de uso del módulo muestra interés*

DOCENCIA10	Añadir muestra de interés
Actor	Administrador y profesor
Descripción	Selecciona y guarda la muestra de interés de una asignatura.
Precondición	El usuario debe tener una cuenta ya registrada y tiene que haberse identificado.
Secuencia normal	1. El sistema presenta las asignaturas de interés. 2. El actor escoge la asignatura por la que se interesa. 3. Finalmente, el actor puede añadir un comentario a la muestra de interés.
Postcondición	El sistema guarda la muestra de interés hecha por el profesor y queda pendiente de lo que decida el administrador.

DOCENCIA11	Eliminar muestra de interés
Actor	Administrador y profesor
Descripción	Selecciona y borra la muestra de interés de una asignatura.
Precondición	El profesor debe de haber mostrado interés por alguna asignatura para poder eliminarla.
Secuencia normal	1. El sistema presenta las asignaturas interesantes para el actor. 2. El profesor escoge la que quiere eliminar.
Postcondición	El sistema elimina el interés mostrado por el profesor a la asignatura seleccionada.
Excepciones	En caso de que la muestra de interés esté tramitada, es decir, que el administrador haya decidido si la acepta o la rechaza, no podrá ser eliminada.

DOCENCIA12	Ver muestras de interés
Actor	Administrador y profesor
Descripción	El sistema presenta las muestras de interés del profesor que ha iniciado sesión.
Precondición	El actor debe tener una cuenta ya registrada y tiene que haberse identificado. Además, debe haber mostrado interés por alguna asignatura.
Secuencia normal	1. El sistema presenta las muestras de interés del actor.
Postcondición	Devuelve las muestras de interés del actor.

DOCENCIA07	Aceptar muestra de interés
Actor	Administrador
Descripción	Acepta una muestra de interés hecha por un profesor y se le asigna el grupo.
Precondición	Deben de haber mostrado interés por alguna asignatura los profesores.
Secuencia normal	1. El sistema presenta las muestras de interés de un profesor. 2. El administrador escoge la muestra que quiere aceptar. 3. En caso de que el número de reservas sea igual o mayor al número de grupos de la asignatura, el administrador le asigna un grupo. En caso contrario, será el profesor en su elección cuando lo decida.
Postcondición	El sistema guarda la reserva confirmada y se asigna la reserva a su elección docente.
Excepciones	En caso de que no queden grupos de la asignatura, directamente marcará la muestra de interés como rechazada.

DOCENCIA08	Rechazar muestra de interés
Actor	Administrador
Descripción	Rechaza una reserva hecha por un profesor.
Precondición	Deben de haber realizado alguna reserva los profesores.
Secuencia normal	1. El sistema presenta las asignaturas reservadas. 2. El administrador escoge la reserva que quiere rechazar.
Postcondición	El sistema elimina la reserva realizada por el profesor.

DOCENCIA09	Deshacer
Actor	Administrador
Descripción	Deshace la decisión tomada sobre una reserva hecha por un profesor.
Precondición	Deben de haberse decidido alguna reserva.
Secuencia normal	1. El sistema presenta las asignaturas reservadas. 2. El administrador escoge la reserva decidida que quiere deshacer.
Postcondición	El sistema deshace la decisión impuesta sobre una reserva.

Capítulo 6

Modelo de datos

En este capítulo se mostrará el modelo de datos utilizado en la implementación de la aplicación.

6.1. Modelo Entidad-Relación

En la figura 6.1 se muestra el diagrama E-R de la aplicación. Se han definido las siguientes entidades:

- Histórico que representa los registros de asignaturas que han impartido los profesores en los últimos años.
- Asignatura que representa a las asignaturas registradas en la aplicación.
- Profesor que simboliza a los profesores registrados.
- ReservaDocencia, representa a los registros de las reservas realizadas por los profesores.
- ReservaDocenciaConfirmada modelizan a las reservas que han sido aceptadas.
- ReservaDocenciaConfirmadaDivisible representa a las reservas aceptadas por el administrador cuya asignatura sea una asignatura divisible.
- MuestraInteres simboliza las muestras de interés que los profesores.

- MuestraInteresConfirmada representa las muestras de interés aceptadas por el administrador.
- MuestraInteresConfirmadaDivisible interpreta a los registros de las reservas aceptadas en las que la asignatura es divisible.

En el mismo diagrama de la figura 6.1 se muestran las relaciones entre las entidades definidas.

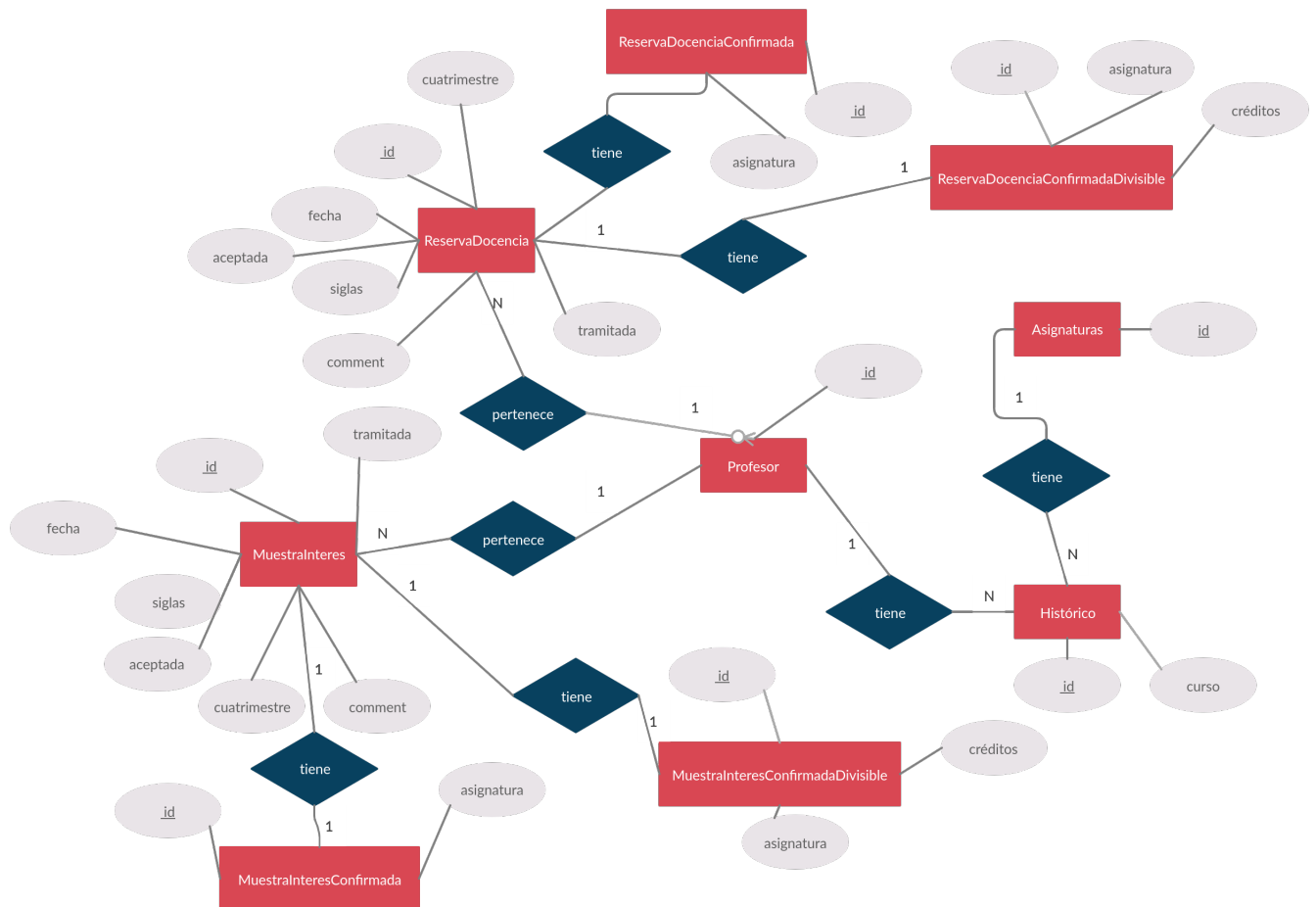


Figura 6.1: Diagrama entidad-relación

6.2. Implementación

El modelo E-R descrito en la sección anterior se ha implementado mediante una base de datos relacional. Para ello, se ha utilizado el modelo de datos Django. Se trata de un *framework* que aporta un acceso muy sencillo a la base de datos mediante una API, ofreciendo operaciones CRUD como método de comunicación principal.

En la figura 6.2 se observan las tablas y como se encuentran relacionadas.

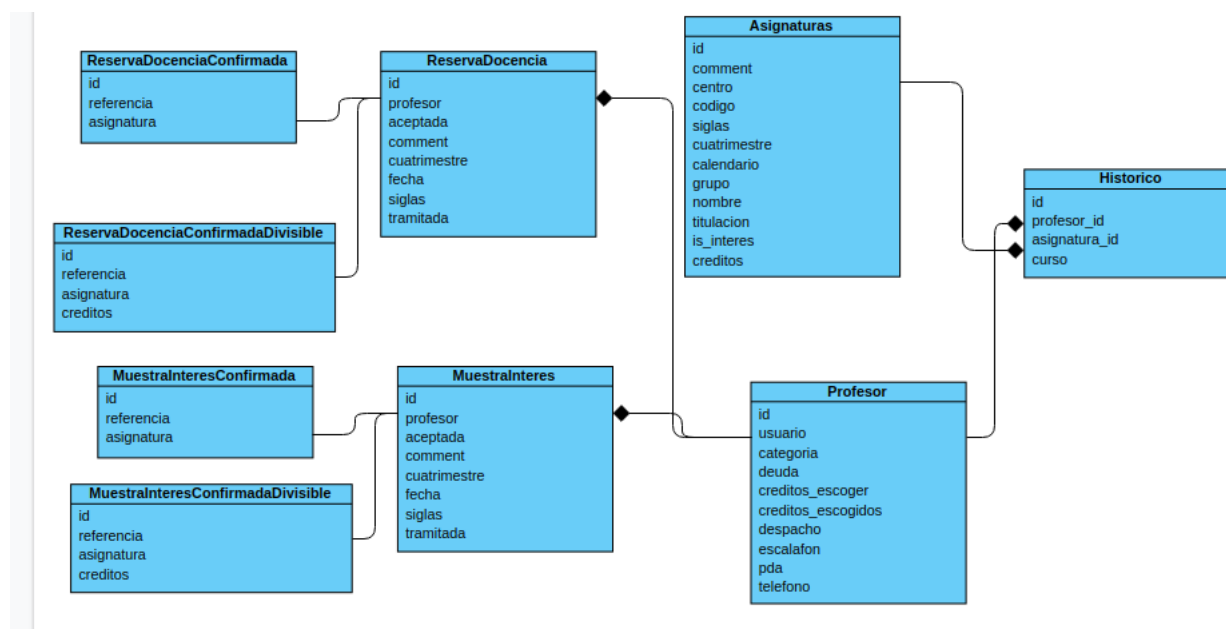


Figura 6.2: Diagrama de clases del proyecto

A continuación detallaremos las tablas y sus atributos exponiéndolas por módulos: histórico, gestión de reservas y gestión de muestras de interés.

6.2.1. Histórico

En esta sección se detallan las tablas de información referente a los históricos registrados. Dispone de los siguientes atributos:

1. La tabla `historico_historico` almacena la información relacionada con el histórico de la aplicación:

- `asignatura_id`: Identificador de asignatura que actúa como clave foránea de la asignatura asociada.
 - `profesor_id`: Identificador de profesor que actúa como clave foránea del profesor asociado.
 - `curso`: Año académico en el que se impartió la asignatura referida por el profesor nombrado.
2. La tabla `historico_importer` es el nombre de la tabla que tiene como función almacenar los documentos importados al histórico. El único dato que alberga es la ruta del archivo.

6.2.2. Gestión de reservas

En esta sección, se explicarán las tablas utilizadas para gestionar las muestras de interés.

1. La tabla `reserva_docencia_reservadocencia` guarda las reservas realizadas por los profesores. Dispone de la siguiente información:
- `fecha`: Cadena que almacena la fecha en la que se realizó la reserva por parte del profesor.
 - `tramitada`: Booleano que indica si el administrador ha decidido sobre la reserva en cuestión.
 - `siglas`: Cadena que indica las siglas de la asignatura reservada.
 - `aceptada`: Booleano que marca si la reserva ha sido aprobada.
 - `profesor_id`: Identificador del profesor que ha solicitado la reserva que actúa como clave foránea.
 - `cuatrimestre`: Es el número que indica el cuatrimestre al que pertenece la asignatura.

- *comment*: Cadena que alberga el comentario que el profesor puede añadir a la reserva.
2. La tabla `reserva_docencia_reservadocenciaconfirmada` se encarga de guardar la información de las reservas aceptadas por el administrador. A continuación se detallarán sus atributos.
- *asignatura*: Es el entero que representa el id de la asignatura reservada y elegida.
 - *referencia_id*: Identificador que representa la reserva asociada a la tupla y que actúa como clave foránea de la reserva asociada.
3. La tabla `reserva_docencia_reservadocenciaconfirmadadivisible` se encarga de almacenar la información de las reservas aceptadas por el administrador en caso de que la asignatura reservada sea divisible. Dispone de los siguientes campos de información:
- *asignatura*: Es el entero que representa el id de la asignatura reservada y elegida.
 - *referencia_id*: Identificador de reserva que actúa como clave foránea de la reserva asociada.
 - *creditos*: Es el número que indica cuantos créditos se han reservado de la asignatura asociada.

6.2.3. Gestión de muestras de interés

En esta sección, se describirán las tablas utilizadas para gestionar la funcionalidad de las reservas.

1. La tabla `muestra_interes_muestraintereres` guarda todas las muestras de interés hacia una asignatura que proponen los profesores. Dispone de la siguiente información:
- *fecha*: Cadena que almacena la fecha en la que se mostró interés por una asignatura.

- tramitada: Booleano que indica si el administrador ha tomado una decisión respecto a la muestra de interés.
- siglas: Cadena que indica las siglas de la asignatura por la que se muestra interes.
- aceptada: Booleano que marca si la muestra de interés ha sido aceptada.
- profesor_id: Identificador de profesor que actúa como clave foránea del profesor que ha mostrado interés.
- cuatrimestre: Es el entero que guarda el cuatrimestre al que pertenece la asignatura.
- comment: Cadena que alberga el comentario que el profesor puede añadir a la muestra de interés.

2. La tabla `muestra_interes_muestrainteresconfirmada` se encarga de guardar la información de las muestras de interés aceptadas por el administrador. Los datos que almacena esta tabla son los siguientes:

- asignatura: Es el número que representa el id de la asignatura elegida.
- referencia_id: Identificador de muestra de interés que actúa como clave foránea de la muestra de interés asociada.

3. La tabla `muestra_interes_muestrainteresconfirmadadivisible` alberga los datos de las muestras de interés aceptadas por el administrador en caso de que la asignatura elegida sea divisible. Los atributos que maneja esta tabla son:

- asignatura: Es el entero que representa el id de la asignatura elegida.
- referencia_id: Identificador de muestra de interés que actúa como clave foránea de la reserva asociada.
- credits: Es el número que indica cuantos créditos se han asignado de la asignatura asociada.

Capítulo 7

Arquitectura de la aplicación

En este capítulo se explicará brevemente la arquitectura de la aplicación. El sistema utiliza un modelo cliente-servidor. Es un modelo de diseño de software en el que las tareas a realizar por la ejecución de la aplicación se reparten entre el cliente y los servidores [19]. En este tipo de arquitecturas es el cliente quien realiza las peticiones al servidor y éste responde con la información adecuada. Una de las características fundamentales de este modelo es la centralización de la gestión de la información, que simplifica mucho el diseño.

Se ha implementado una interfaz de programación de aplicaciones (API) [20]. Los motivos de esta decisión han sido la facilidad con la que se acceden a los recursos y a la vez, la seguridad y el control que otorga. Además, tiene gran capacidad de intercambio de información entre múltiples plataformas.

En la figura 7.1 se representa el diseño de la arquitectura del sistema. Tras la compilación de los componentes del proyecto Angular, el resultado es código HTML y Javascript que ejecutará el cliente de la aplicación en el navegador. Las peticiones se hacen mediante HTTP y su destino son los servicios de la API. Ésta, tras procesarlas, devuelve los datos solicitados conforme a la información que se halla en la base de datos y finalmente acaba llegando de vuelta al cliente.

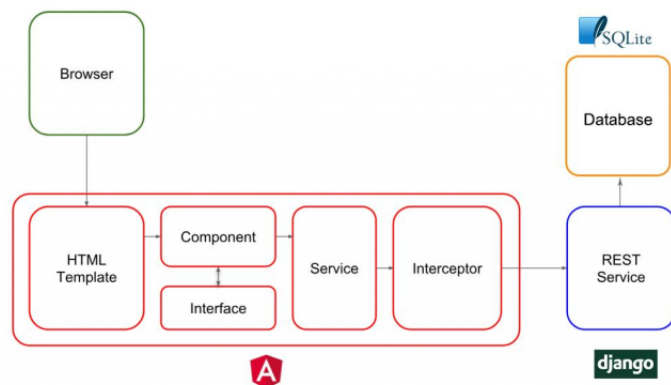


Figura 7.1: Esquema representativo de la arquitectura del sistema[1]

7.1. Patrones de diseño

En este apartado se describirán los patrones de diseño de los *frameworks* implementados en el proyecto.

7.1.1. Capa de presentación

La capa de presentación sigue el patrón de diseño modelo-vista-vista-modelo (MVVM) [9]. Esto se debe a una propiedad fundamental de Angular, *two way data binding*. Este concepto permite modificar cualquier elemento en la vista y que quede reflejado en el modelo y viceversa. Esto facilita la omisión del controlador en la arquitectura de la web, que consigue simplificar el desarrollo. También existen los servicios, lógicas de negocio que se encargan de las llamadas a la API.

7.1.2. Servicio REST

El servicio REST implementado se llama Django [15]. Al igual que ha pasado con la capa de presentación, Django tampoco sigue el patrón MVC, si no que utiliza el patrón MTV (modelo-plantilla-vista). En él, la vista es la capa de negocios, es decir, se puede considerar un puente entre el modelo y la plantilla. Además, se apoya en las plantillas para decidir la presentación los datos requeridos. La API está conectada con una base de datos relacional

SQLite, que se encuentra en el mismo servidor que el servicio.

Capítulo 8

Diseño de la aplicación

En este capítulo se detallará la implementación de todas las funcionalidades desarrolladas en este proyecto.

Al tratarse de una ampliación de un sistema de elección de docencia, el diseño de la aplicación ya estaba decidido. Es un diseño simple e intuitivo, que tiene como objetivo albergar todas las funcionalidades en una misma vista. Para conseguirlo, se han seguido las pautas de diseño de Material Design [21].

Para la capa de presentación, se ha utilizado Angular. Se trata de un *framework* que, debido a su naturaleza modular, facilita mucho el desarrollo del sistema. Además, ofrece una sencilla comunicación entre el *frontend* y la API.

8.1. Funcionalidad de la aplicación

En esta sección se describirán las funcionalidades implementadas divididas en módulos: histórico, gestión de las reservas, gestión de las muestras de interés y elección docencia.

8.1.1. Histórico

El módulo histórico se ocupa de almacenar las elecciones de docencia del profesorado del departamento. Su objetivo como funcionalidad es dar soporte a la gestión de las reservas. En la figura 8.1 se muestra la pantalla principal del histórico. En ella se encuentran y se desarrollan prácticamente todas las funcionalidades que alberga.

La vista se divide en dos partes. La primera, que se encuentra en el centro de la pantalla, es la tabla donde se muestran los históricos. La segunda, a la derecha, se trata de un bloque con botones y paneles extensibles que reúnen las funcionalidades del módulo.

Para facilitar observar los datos que interesan sobre los históricos buscados, existe el panel extensible 'Mostrar' que se encarga de mostrar/ocultar las columnas que se deseen.



The screenshot shows a web interface titled 'Histórico'. It features a table with four columns: 'Profesor', 'Siglas/Asignatura', 'Año académico', and 'Curso'. The table contains 12 rows of data. To the right of the table is a sidebar with a 'Filtrar' dropdown, two buttons 'Importar histórico' and 'Exportar historicos', and a 'Mostrar' panel. The 'Mostrar' panel has checkboxes for 'Profesor', 'Código', 'Nombre', 'Curso', 'Año académico', 'Grupo', 'Cuatrimestre', 'Titulación', and 'Departamento'. The 'Curso' and 'Año académico' checkboxes are checked.

Profesor	Siglas/Asignatura	Año académico	Curso
Narciso Martí Oliet	[TP2] Tecnología de la programación 2	2017/2018	2º
Narciso Martí Oliet	[TDM] Tratamiento de datos masivos	2017/2018	
Narciso Martí Oliet	[MAR2] Métodos algorítmicos en resolución de problemas 2	2017/2018	3º
Francisco J. López Fraguas	[MUS] Informática musical	2017/2018	Opt.
Francisco J. López Fraguas	[SGD] Sistemas de Gestión de Datos y de la Información	2017/2018	1º
Francisco J. López Fraguas	[DA] Diseño de algoritmos	2017/2018	3º
Francisco J. López Fraguas	[BIGD] Datos masivos: Big Data	2017/2018	Opt.
Francisco J. López Fraguas	[PD] Programación Declarativa	2017/2018	4º
Francisco J. López Fraguas	[MAR2] Métodos algorítmicos en resolución de problemas 2	2017/2018	3º
Manuel Núñez García	[DA] Diseño de algoritmos	2017/2018	3º
Manuel Núñez García	[MAR2] Métodos algorítmicos en resolución de problemas 2	2017/2018	3º

Figura 8.1: *Lista de los históricos*

El panel 'Filtrar' guarda en su interior un filtro de búsqueda. Las opciones que ofrece la funcionalidad se pueden ver en la figura 8.2. En la tabla no se mostrará ningún dato almacenado mientras no se haga ninguna consulta en el formulario. Esta búsqueda solo permitirá utilizar el *combobox* del profesor cuando el actor sea el administrador.

Para conseguir los resultados de la búsqueda, la aplicación utiliza llamadas a la API. Es por ello, que la vista lo único que hace es observar los cambios de los parámetros de búsqueda, enviar la información y actualizar la tabla con la nueva información. En la figura 8.3 se muestran los métodos que dan solución a este problema.

Panel de filtros de los históricos. El panel contiene los siguientes elementos:

- Título: Filtrar
- Selección de profesor: "Seleccione un profesor"
- Selección de año: "Seleccione un año"
- Selección de siglas: "Siglas"
- Selección de curso: "Curso"
- Selección de cuatrimestre: "Cuatrimestre"
- Botón: Clear
- Icono de lupa (búsqueda)
- Botón: Importar histórico
- Botón: Exportar históricos
- Botón: Mostrar

Figura 8.2: Panel de filtros de los históricos

```

162
163 buscarHistorico(): void {
164     this.loading = true;
165
166     if(!this.admin) {
167         this.searchVals.profesor_id =
168             this.profesor.usuario.id.toString();
169     }
170     this.historicoService.searchHistorico(
171         this.searchVals.profesor_id,
172         this.searchVals.anyo,
173         this.searchVals.siglas,
174         this.searchVals.curso,
175         this.searchVals.cuatrimestre
176     ).subscribe(data =>{
177         this.arrayHistoricoAux = data;
178         this.loading = false;
179     });
180 }
181
182
41
42 getHistoricos(): Observable<Historico[]> {
43     return this.http.get<Historico[]>(this.historicoUrl
44 )
45 }
46
47 searchHistorico(profesor_id: string, curso: string,
48 siglas: string,
49 cursoAsignatura: string,
50 cuatrimestre: string): Observable<Historico[]> {
51     var params = 'profesor_id=' + profesor_id +
52         '&curso=' + curso +
53         '&cuatrimestre=' + cuatrimestre + '&anyo=' +
54         cursoAsignatura + '&siglas=' + siglas;
55
56     return this.http.get<Historico[]>(
57         this.historicoUrl + '?' + params);
58 }
59
60 exportToExcel(json: any[], fileName: string): void {
61     const worksheet: XLSX.WorkSheet = XLSX.utils.json_to
62         worksheet(XLSX.WorkBook, f

```

Figura 8.3: Fragmento de código - Búsqueda de históricos

8.1.1.1. Exportar histórico

Se trata de un botón cuya función es exportar a un fichero Excel los datos mostrados en la tabla. Para conseguirlo, hay que pulsar en el botón 'Exportar históricos'. Esta funcionalidad solo está permitida cuando haya alguna búsqueda realizada.

8.1.1.2. Importar histórico

Al pulsar sobre el botón 'Importar histórico', aparecerá una nueva vista. En ésta, se observa un pequeño resumen de cómo funciona y un ejemplo de archivo de importación (ver la figura 8.4).

Los históricos se importan por años. Es decir, los archivos que se subirán a la aplicación abarcan toda la docencia de los profesores de un año académico determinado. Dependiendo de si existen datos de ese año en la base de datos, sobrescribirá los datos existentes en caso positivo, o simplemente añadirá los datos encontrados en el fichero subido. Esta funcionalidad se realiza a nivel de API y es allí donde se ejecuta el control de errores de la funcionalidad, que se mostrarán por pantalla en caso de existir.

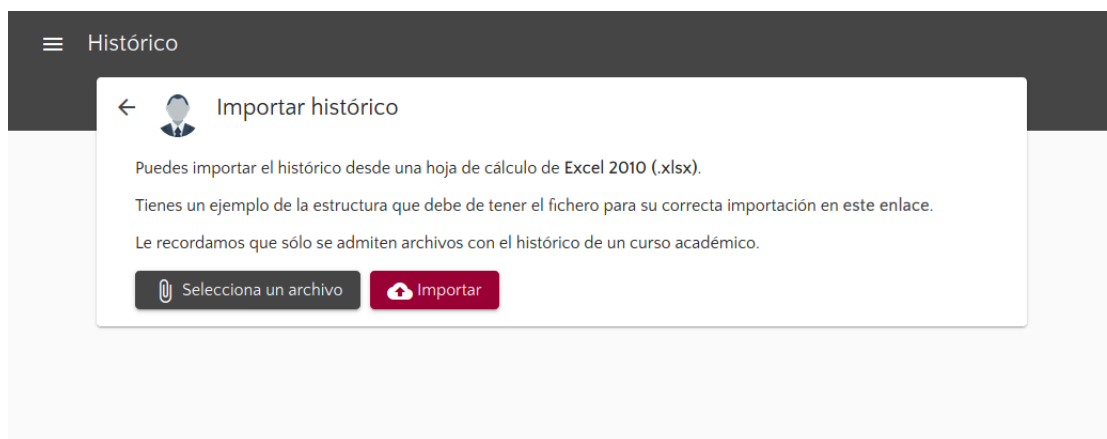


Figura 8.4: *Vista para importar históricos*

8.1.2. Gestión de reservas

La gestión de las reservas es la funcionalidad principal del proyecto. Tiene como objetivo automatizar el proceso de reserva de docencia por parte de los profesores. Este proceso ofrece a los profesores la posibilidad de decidir parte de su elección docencia antes de que comience ésta. No todas las asignaturas se pueden reservar. Para que un profesor tenga la posibilidad de reservar una asignatura, tiene que cumplir una serie de requisitos.

- El profesor debe haber cursado la asignatura el año anterior a la realización de la

reserva.

- La asignatura no debe haberse impartido más de 2 veces en 6 años.
- La asignatura reservada no puede ser de primero o desdoble.

The screenshot shows a web interface titled 'Reserva docencia'. It features a main table with columns: 'Reservar', 'Siglas/Asignatura', 'Curso', and 'Cuatrimestre'. The table lists several courses, some of which are highlighted in grey to indicate they are reserved. A sidebar on the right contains a 'Filtrar' section with dropdowns for 'Siglas', 'Curso', and 'Cuatrimestre', a 'Clear' button, and a search icon. Below the filters is a 'Mis reservas' section showing a list of reserved courses with status icons. At the bottom of the sidebar is a 'Mostrar' dropdown.

Reservar	Siglas/Asignatura	Curso	Cuatrimstre
Añadir	[EDA] Estructuras de datos y algoritmos	2º	1
Eliminar	[TPI] Tecnología de la programación 1	2º	1
Añadir	[GIW] Gestión de la Información en la Web	Opt.	1
Eliminar	[FAL] Fundamentos de algoritmos	2º	1
Resto de asignaturas			
Añadir	[IN] Informática	2º	1
Añadir	[IG1] Informática Gráfica I	2º	2
Añadir	[IG1] Informática Gráfica I	2º	2
Añadir	[IG2] Informática Gráfica II	3º	1
Añadir	[IG2] Informática Gráfica II	3º	1
Añadir	[MUS] Informática musical	Opt.	2
Añadir	[ISI] Ingeniería del software 1	2º	1
Añadir	[ISI] Ingeniería del software 1	2º	1
	[IS2] Ingeniería del software 2	2º	2
Añadir	[IS2] Ingeniería del software 2	2º	2
Añadir	[IPM] Introducción a la Programación Matemática		1
Añadir	[TRF] Introducción a la tecnología blockchain u smart contracts	Opt.	1

Figura 8.5: *Página principal del módulo gestión de reservas*

Esta funcionalidad abarca dos vistas: la común para todos los profesores (ver figura 8.5) y la del administrador, que se verá más adelante. Comenzando por la primera, se observa en la parte central de la pantalla dos tablas con asignaturas. La primera de ellas muestra las asignaturas que cumplen los requisitos para ser reservadas por el profesor.

La primera columna de las tablas tiene como función añadir o eliminar la reserva de la asignatura elegida. En el caso de que se quiera añadir una reserva, saltará un *pop-up* como método de confirmación. El profesor podrá añadir un comentario a la reserva para que el administrador lo lea. En la figura 8.6 se pueden ver las llamadas a la API de ambas funcionalidades: añadir y eliminar.

En caso de estar reservada la asignatura, se mostrará de otro color la fila entera para poder diferenciarla del resto de asignaturas no reservadas. Si el profesor quisiera eliminar

```

293
294 anadirReserva(asignatura: Asignatura, comment: string): void {
295     this.loading = true;
296     this.reserva = new Reserva();
297     this.reserva.siglas = asignatura.siglas;
298     this.reserva.fecha = this.diaHoy.toDateString();
299     this.reserva.profesor = this.profesor.usuario.id;
300     this.reserva.cuatrimestre = asignatura.cuatrimestre;
301     this.reserva.tramitada = false;
302     this.reserva.aceptada = false;
303     if(comment == undefined) comment = null;
304     this.reserva.comment = comment;
305
306     this.reservaService.saveReserva(this.reserva).subscribe(data => { // data is already a JSON object
307         console.log(data);
308         this.reservasHechas.push(data);
309         this.avisosService.enviarMensaje("Se ha guardado la reserva correctamente");
310         this.actualizarAsignaturas(asignatura, true);
311     },
312     err => {
313         this.avisosService.enviarMensaje("Error al crear la reserva, por favor, intentelo de nuevo");
314         this.loading = false;
315     });
316 }
317
318 eliminarReserva(asignatura: Asignatura): void {
319     var eliminado = false;
320     this.reservaService.getReservasHechas(this.profesor.usuario.id).subscribe(
321         data => {
322             data.forEach(r => {
323                 if(!eliminado && r.siglas == asignatura.siglas &&
324                     r.cuatrimestre == asignatura.cuatrimestre) {
325                     eliminado = true;
326                     this.reservaService.deleteReserva(r.id).subscribe(
327                         data => {
328                             this.reservasHechas.splice(this.reservasHechas.indexOf(data), 1);
329                             this.avisosService.enviarMensaje("Se ha eliminado la reserva correctamente");
330                             this.actualizarAsignaturas(asignatura, false);
331                         },
332                         err => {
333                             this.avisosService.enviarMensaje("Error al eliminar la reserva, por favor, intentelo de nuevo");
334                             this.loading = false;
335                         }
336                     );
337                 }
338             });
339         }, err => {
340         }
341     );
342 }
343 }
344

```

Figura 8.6: Fragmento de código - Añadir y eliminar reserva

alguna reserva, bastará con pulsar sobre el botón 'Eliminar' y la reserva se eliminará. Una vez el administrador haya decidido sobre las reservas, no se podrán eliminar, sea cual sea la decisión tomada.

A la derecha de la pantalla, se encuentran 4 cajas de las cuáles 3 son paneles extensibles. El primero de ellos, 'Filtrar', sirve para simplificar la búsqueda de asignaturas reservables. Los campos por los que se pueden filtrar las asignaturas son: siglas, cuatrimestre y curso.

El segundo panel, 'Mis reservas' lista las reservas realizadas por el profesor. Para ello, se muestran las siglas de la asignatura reservada, el cuatrimestre al que pertenece y finalmente dos iconos. El segundo de ellos muestra el comentario que posee la reserva. Al pulsar sobre él, saltará un *pop-up* con la información descrita previamente. El icono que resta marca el estado de la reserva: aceptada (se mostrará un *check*), rechazada (representado con una cruz), o indeciso (simbolizado con un interrogante). En la figura 8.7 se puede apreciar.

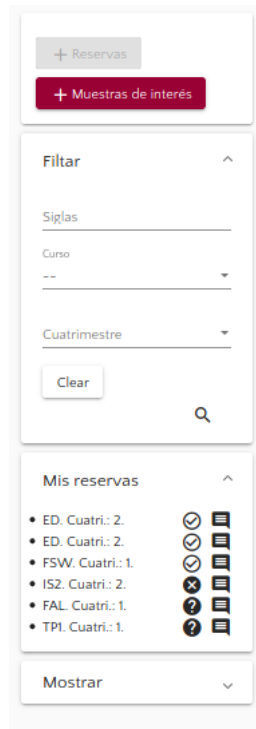


Figura 8.7: *Panel de opciones para las reservas*

El tercer y último panel, 'Mostrar', al igual que en el módulo histórico, tiene como función elegir las columnas que se presentan en las tablas.

La caja que queda por describir contiene dos botones: 'Reservas' y 'Muestras de interés'. Este panel sirve para cambiar las asignaturas que se muestran en las tablas y cambiar de vista a 'Muestra Interés'. Ambos módulos comparten la estructura visual y funcional, por lo que se encuentran en la misma vista. Dependiendo de qué datos esté mostrando la tabla, el botón relacionado con estos datos no estará disponible.

En caso de que el actor sea el administrador, en la parte superior de la pantalla aparecerá un recuadro con un botón 'Confirmar' y un *combobox* con todos los profesores registrados en la aplicación. Si pulsamos en el botón, nos llevará a otra vista, donde el administrador tendrá que decidir sobre las reservas realizadas (ver figura 8.8).


```

toMS(){
  this.loading = true;
  MenuToolBarComponent.updateTitle("Muestra de interés");

  this.isReserva = false;
  this.asignaturas.map(a => a.interested = false);
  //
  this.getAsignaturas();
}
toReservas(){
  this.loading = true;
  MenuToolBarComponent.updateTitle("Reserva docencia");
  this.lastIDAsigReservables = -1;
  this.asignaturasResto = [];
  this.asignaturasRestoAux = [];
  this.asignaturasAux = [];
  this.isReserva = true;
  if(this.opts.grupo){
    this.updateNumCols(this.opts.grupo,1);
    this.opts.grupo = false;
  }
  this.asignaturas.map(a => a.interested = false);
  //
  this.getAsignaturas();
}

```

Figura 8.8: *Métodos encargados del cambio de datos entre reserva y muestra de interés*

8.1.2.1. Vista administrador

La composición de pantalla es muy similar a la vista anterior, una tabla en el centro de la pantalla y dos cajas en la parte derecha. La tabla mostrará las reservas realizadas por el profesor indicado en el *combobox* de la parte superior. Dependiendo del estado de la reserva, se mostrarán diferentes iconos en la columna 'Confirmar' y la fila de la reserva en cuestión tendrá un color u otro (visible en la figura 8.9).

En caso de no haber decidido sobre una reserva, se mostrará en blanco y con los botones para decidir sobre ella. Con independencia de lo que decida, saltará un *pop-up* para comunicar al administrador si quiere aportar algún comentario a la reserva para que el profesor pueda leerlo y entender por qué se le ha concedido o no. Si la decisión tomada sobre la reserva es aceptar, aparecerá un formulario como el de la figura 8.10 si el número de reservas a esa asignatura es igual o mayor que el número de grupos que existen de ésta. Si por el contrario es menor, se dejará la elección del grupo para el proceso de elección de docencia del profesor.

Reserva docencia			
← Profesor: David de Fruto... ▾			
Siglas	Cuatri	Comentario	Confirmar
ED	2		⌂
ED	2		⌂
FSW	1		⌂
IS2	2		⌂
FAL	1		✓ ✕
GBD	1		✓ ✕
PCOM	1		✓ ✕

+ Reservas
+ Muestras de interés
Alertas (3) ▾

Figura 8.9: *Lista de reservas aceptadas*

Aviso

¿Estás seguro/a que quieres realizar esta acción?
El número de reservas de esta asignatura es igual o superior al número de grupos que ésta tiene, por lo que tendrá que elegir el grupo el administrador. Si no ha hablado con el profesor que ha realizado la reserva, hágalo antes de continuar.

Elige grupo y horario
Grupo: - Horario: M: 14:00:00-15:50:00/J: 14:00:00-15:50:00/ ▾

Creditos seleccionados 10

Añade algún comentario

Cancelar Confirmar

Figura 8.10: *Formulario de decisión de una reserva*

Si la reserva ya está resuelta, el administrador siempre podrá deshacer la decisión tomada mediante el botón que aparece en la columna 'Confirmar'. En la figura 8.11 se puede observar un fragmento de código de esta funcionalidad.

Cabe la posibilidad de que el número de grupos disponibles de una asignatura (grupos sin reserva ni docencia) se reduzca a 0 durante el proceso de elección. En tal caso, si existe alguna reserva aceptada sobre la asignatura, será el administrador quien empareje los grupos

```

172 undoObj(obj, MuestraInteres[Reserva]: void)
173 this.loading = true;
174 this.isReserva ? this.undoObjConf = new ReservaConfirmada(); this.undoObjConf = new MuestraInteresConfirmada();
175 we anulado = false;
176 if(this.isReserva) {
177   this.undoObjConf = this.reservasConf.find(r=r.r.referencia == obj.id);
178   if(this.undoObjConf == undefined) {
179     delete this.undoObjConf;
180     this.undoObjConfDivisible = new ReservaConfirmadaDivisible();
181     this.undoObjConfDivisible = this.reservasConfDivisible.find(r=r.r.referencia == obj.id);
182   }
183 } else {
184   this.undoObjConf = this.muestrasConf.find(r=r.r.referencia == obj.id);
185   if(this.undoObjConf == undefined) {
186     delete this.undoObjConf;
187     this.undoObjConfDivisible = new MuestraInteresConfirmadaDivisible();
188     this.undoObjConfDivisible = this.muestrasConfDivisible.find(r=r.r.referencia == obj.id);
189   }
190 }
191 console.log(this.undoObjConf);
192 console.log(this.undoObjConfDivisible);
193 if(obj.aceptada && ((this.undoObjConf != undefined && this.undoObjConf.asegnatura != -1) || this.undoObjConfDivisible != undefined)){
194   this.eleccionService.getEleccionProfesor(this.profesor.usuario.id).subscribe(
195     eleccion => {
196       if(eleccion.length != 0){
197         if(this.eleccion != undefined) delete this.eleccion;
198         this.eleccion = new Eleccion();
199         this.eleccionAntigua = eleccion[0];
200         console.log(this.eleccionAntigua);
201         this.eleccionService.deleteEleccion(this.eleccionAntigua.id).subscribe(data => {
202           if(this.undoObjConf == undefined) {
203             const ad = this.eleccionAntigua.asegnaturas_divisibles.find(aseg => aseg.asegnatura.id == this.undoObjConfDivisible.asegnatura);
204             this.eleccionAntigua.asegnaturas_divisibles.splice(this.eleccionAntigua.asegnaturas_divisibles.indexOf(ad), 1);
205           } else {
206             const a = this.eleccionAntigua.asegnaturas.find(aseg => aseg.id == this.undoObjConf.asegnatura);
207             this.eleccionAntigua.asegnaturas.splice(this.eleccionAntigua.asegnaturas.indexOf(a), 1);
208           }
209         });
210         //this.deleteConfirmadas(obj);
211         if(this.eleccionAntigua.asegnaturas.length != 0 || this.eleccionAntigua.asegnaturas_divisibles.length != 0){
212           this.eleccion.asegnaturas = this.eleccionAntigua.asegnaturas;
213           this.eleccion.asegnaturas_divisibles = this.eleccionAntigua.asegnaturas_divisibles;
214           this.eleccion.deudables = this.eleccionAntigua.deudables;
215           this.eleccion.deuda = this.eleccionAntigua.deuda;
216           this.eleccion.profesor = this.eleccionAntigua.profesor;
217           this.eleccionService.createEleccion(this.eleccion).subscribe(()=>{
218             console.log('createEleccion bien');
219             this.deleteConfirmadas(obj);
220             obj.tramitada = false;
221             obj.aceptada = false;
222             this.actualizarObj(obj);
223           });
224         } else {
225           this.deleteConfirmadas(obj);
226           obj.tramitada = false;
227           obj.aceptada = false;
228           this.actualizarObj(obj);
229         }
230       }
231     }
232   );
233 }

```

Figura 8.11: *Fragmento de código - Deshacer reserva*

restantes a las reservas. Para ello, aparecerá un nuevo botón de confirmar a la izquierda del botón 'Deshacer' de la reserva.

A la derecha de la pantalla se encuentran las alertas y la caja descrita previamente para cambiar entre reservas y muestras de interés. El panel extensible 'Alertas' recoge la función de avisar al administrador de las asignaturas, mostrando las siglas y el cuatrimestre de la asignatura reservada, que necesitan su decisión. Esta situación se produce cuando el número de grupos disponibles de la asignatura es 0 (visible en la figura 8.12).



Figura 8.12: *Alertas de reservas pendientes*

8.1.3. Gestión de muestras de interés

Otro proceso que se involucra en el proceso de elección de docencia son las muestras de interés. Existen ciertas asignaturas que se consideran 'especiales' y por las que cualquier profesor puede mostrar interés. Éstas son las pertenecientes a los máster, asignaturas impartidas en inglés y las optativas.

Muestra de interés

Interés	Siglas/Asignatura	Curso	Cuatrimestre
Añadir	[MDL1] Matemática Discreta y Lógica Matemática 1	1º	1
Eliminar	[MDL2] Matemática Discreta y Lógica Matemática	1º	2
Añadir	[TP1] Tecnología de la programación 1	2º	1
Añadir	[TP2] Tecnología de la programación 2	2º	2
Añadir	[BD] Bases de datos	2º	1
Eliminar	[NSQ] Bases de datos NoSQL	Opt.	2
Añadir	[MIN] Minería de datos y el paradigma Big Data	Opt.	2
Añadir	[GIW] Gestión de la Información en la Web	Opt.	1
Eliminar	[PR] Programación con restricciones	Opt.	2
Añadir	[TSW] Testing de software	Opt.	1

+ Reservas

+ Muestras de interés

Filtrar

Mis intereses

Mostrar

☒ Nombre

☒ Curso

☐ Grupo

☒ Cuatrimestre

☐ Departamento

Figura 8.13: Listado de las asignaturas de interés.

Las funcionalidades son exactamente iguales que las del módulo de las reservas. En lo que respecta a la vista, es prácticamente igual salvo por el detalle de que en este módulo solo existe una tabla de asignaturas por las que mostrar interés. Para acceder a este módulo, se deberá pulsar sobre el botón de la pantalla superior derecha 'Muestra de interés' (ver en la figura 8.13). Se mostrarán todas las asignaturas por las que un profesor puede interesarse. Al igual que en la anterior sección, se pueden añadir y eliminar muestras de interés, que se verán reflejadas en el panel extensible de la derecha 'Mis intereses' (ver figura 8.14). Dispone de los mismos filtros que la gestión de las reservas. La única diferencia que muestra esta vista con el módulo anterior es que en el panel 'Mostrar' se puede añadir la columna 'Grupo', dado que las muestras de interés son hacia asignaturas concretas.

```

396   });
397 }
398 save(asignatura: Asignatura, comment: string): void {
399   this.loading = true;
400   this.muestraInteres = new MuestraInteres();
401   this.muestraInteres.siglas = asignatura.siglas;
402   this.muestraInteres.fecha = this.diaHoy.toDateString();
403   this.muestraInteres.profesor = this.profesor.usuario.id;
404   this.muestraInteres.cuatrimestre = asignatura.cuatrimestre;
405   this.muestraInteres.tramitada = false;
406   this.muestraInteres.aceptada = false;
407   if(comment == undefined) comment = null;
408   this.muestraInteres.comment = comment;
409
410   this.interesService.saveMuestraInteres(this.muestraInteres).subscribe(
411     data => { // data is already a JSON object
412       this.interesHechas.push(data);
413       this.avisosService.enviarMensaje("Se ha creado la muestra correctamente");
414       this.actualizarAsignaturas(asignatura, true);
415       //this.getAsignaturas();
416     },
417     err => {
418       console.log(err);
419       this.avisosService.enviarMensaje("Error al crear la muestra, por favor, revisa el formulario");
420       this.loading = false;
421     }
422   );
423 }
424 eliminarInteres(asignatura: Asignatura): void {
425   var eliminado = false;
426   this.interesService.getMuestrasInteresProfesor(this.profesor.usuario.id).subscribe(
427     intereses => {
428       intereses.forEach(i => {
429         if(!eliminado && i.siglas == asignatura.siglas &&
430           i.cuatrimestre == asignatura.cuatrimestre) {
431           eliminado = true;
432           this.interesService.deleteInteres(i.id).subscribe(
433             data => {
434               this.interesHechas.splice(this.interesHechas.indexOf(data), 1);
435               this.avisosService.enviarMensaje("Muestra de interés eliminada correctamente");
436               this.actualizarAsignaturas(asignatura, false);
437               //this.getAsignaturas();
438             },
439             err => {
440               this.avisosService.enviarMensaje("Muestra de interés no se ha podido eliminar, intentelo de nuevo");
441               this.loading = false;
442             }
443           );
444         }
445       });
446     },
447     err => {
448       console.log("error al buscar muestras de profesor");
449     }
450   );
451 }
452 }

```

Figura 8.14: Fragmento de código - Añadir y eliminar muestras de interés

8.1.3.1. Vista administrador

En esta sección, el funcionamiento sigue siendo idéntico. La única diferencia es que en este módulo, no existen las reelecciones, es decir, la asignaturas por las que pueden mostrarse interés son de un solo grupo y por consiguiente, el administrador siempre decidirá sobre los intereses (ver en la figura 8.15).

El panel 'Alertas' es general para ambas vistas, por lo que su funcionamiento no varía.

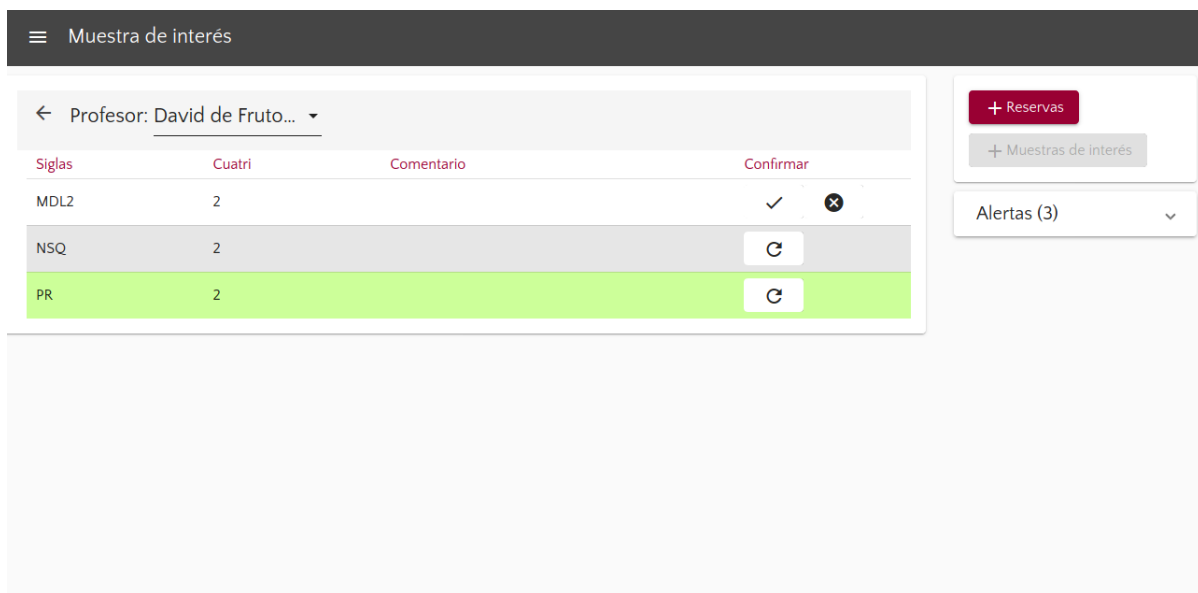


Figura 8.15: *Lista de muestras de interés solicitados por un profesor*

8.1.4. Elección de docencia

Este módulo ya estaba desarrollado en el proyecto. Se ha actualizado para que se tuvieran en cuenta las reservas y las muestras de interés. La actualización tiene un objetivo muy sencillo, tener en cuenta las reservas y las muestras de interés aceptadas de los profesores y las muestras de interés a la hora de realizar la elección docente.

Para ello, existe un método que detecta si la asignatura seleccionada en la elección pertenece a alguna de las reservas del actor (ver figura 8.16). Este control también se realiza para las asignaturas divisibles, que junto con las normales, son las únicas que se pueden reservar.

A la hora de finalizar la elección, en caso de que el profesor no haya seleccionado un grupo de las reservas que tiene, dicha reserva pasará a estar en el estado no aceptada (ver figura 8.17).

Capítulo 9

Conclusiones y trabajo futuro

9.1. Conclusiones

En este trabajo se ha continuado el desarrollo de un sistema de elección de docencia [1]. El objetivo es facilitar la gestión de la elección de docencia de un departamento universitario. En ese aspecto, se ha conseguido completar un poco más la aplicación, se ha permitido a la herramienta almacenar las elecciones de cursos pasados y además se ha automatizado la gestión de reservas y muestras de interés de ciertas asignaturas, procesos que se utilizan actualmente en el departamento.

Gracias a la aplicación los profesores disponen de un soporte donde podrán elegir docencia y consultar los horarios en el calendario hasta quedarse con el que más les convenga. El administrador también saldrá beneficiado, ya que dispondrá de un sistema donde se almacenará gran cantidad de la información que suele manejar, el histórico para las reservas y un apoyo en todo el proceso de la elección de docencia.

9.2. Trabajo futuro

A continuación se expondrán las líneas de trabajo futuro para completar este sistema:

- **Sistema de autenticación.** Cada profesor que quiera utilizar este sistema deberá estar registrado, lo que conlleva tener unas credenciales específicas para la aplicación. El objetivo es que los profesores utilicen las credenciales de la intranet de la facultad

y así conseguir una única credencial.

- **Gestión de personal para el cuidado de exámenes.** Esta funcionalidad no está relacionada con la elección de docencia, pero sería de gran utilidad para el departamento la digitalización de este proceso.
- **Mejoras y correcciones.** Conforme se vaya utilizando la aplicación, surgirán problemas de funcionamiento o ciertas partes del desarrollo quedarán obsoletas. Es por ello, que el mantenimiento de la aplicación es un pilar fundamental en lo que respecta a su desarrollo.

Capítulo 10

Conclusions and future work

10.1. Conclusions

In this work, the development of a teaching choice system has continued. The objective is to facilitate the management of the choice of a university department. In this respect, the application has been completed a little more, the tool has been allowed to store choices of past courses and the management of reserves and samples of interest of certain subjects has been automated, processes currently in use in the department.

Teachers have a support where they can try out different choices to stay with the one that suits them, as well as be able to reserve and show interest in subjects. On the other hand, the administrator will have a system where a lot of the information that it usually handles will be stored.

Thanks to the application the teachers have a support where they can choose teaching and consult the timetable in the calendar until they stay with the one that suits them. The administrator will also benefit, he will have a system where it will store a lot of the information he usually handles, the history for the reserves and support in the entire process of the teaching choice.

10.2. Future work

The following are the future lines of work to complete this system:

- **Authentication system.** Every teacher who wants to use this system must be registered, which entails having specific credentials for the application. The aim is for teachers to use the faculty's intranet credentials, thus obtaining a single credential.
- **Personnel management for exam care.** This functionality is not related to the choice of teaching, but it would be very useful for the department to digitize this process.
- **Improvements and fixes.** As the application is used, operational problems will arise or certain parts of the development will remain

Bibliografía

- [1] Arroyo Segovia, D., Picatoste Zangróniz, J. (2019). Desarrollo de un sistema de gestión de la elección de docencia.
- [2] «Bullet Education Scheduling and Timetabling». [En línea]. Disponible en: <https://www.bulletsolutions.com/es/best-bullet-education-scheduling/>
- [3] «AscHorarios». [En línea]. Disponible en: https://www.asctimetables.com/timetables_es.html
- [4] «OpenERP Odoo». [En línea]. Disponible en: https://www.odoo.com/es_ES/page/education-program
- [5] «Fedena». [En línea]. Disponible en: <https://fedena.com/>
- [6] «Visual Studio Code». [En línea]. Disponible en: https://es.wikipedia.org/wiki/Visual_Studio_Code
- [7] «Node Package Manager». [En línea]. Disponible en: [https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))
- [8] «Angular». [En línea]. Disponible en: <https://angular.io/>
- [9] «Model–view–viewmodel - Implementations». [En línea]. Disponible en: <https://en.wikipedia.org/wiki/Model\T1\textendashview\T1\textendashviewmodel>
- [10] «HTML5». [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/HTML/HTML5>
- [11] «HTML». [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTML>
- [12] «CSS3». [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>
- [13] «Sass documentation». [En línea]. Disponible en: <https://sass-lang.com/>
- [14] «Apache HTTP Server». [En línea]. Disponible en: https://en.wikipedia.org/wiki/Apache_HTTP_Server

- [15] «What Is SQLite?» [En línea]. Disponible en: <https://www.sqlite.org/draft/index.html>
- [16] «Django REST framework». [En línea]. Disponible en: <https://www.django-rest-framework.org/>
- [17] «Python Virtual Environment». [En línea]. Disponible en: https://wiki.archlinux.org/index.php/Python/Virtual_environment
- [18] «Librería openpyxl». [En línea]. Disponible en: <https://openpyxl.readthedocs.io/en/stable/>
- [19] «El modelo cliente-servidor». [En línea]. Disponible en: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>
- [20] «Client-server model». [En línea]. Disponible en: <https://es.wikipedia.org/wiki/Cliente-servidor>
- [21] «Material Design». [En línea]. Disponible en: <https://material.io/>

Apéndice A

Guía de uso

En este primer apéndice, se detallarán las funcionalidades desarrolladas en este proyecto. Para cada una de ellas, primero se mostrará el funcionamiento cotidiano y común a todos los profesores. Tras esto y si existe alguna particularidad en el funcionamiento de la aplicación con las credenciales de administrador, se detallarán los eventos sucedidos.

A.1. Consultar histórico

Para consultar en el histórico, hay que ubicarse en la sección 'Histórico'. Una vez allí, en la parte derecha de la pantalla se encuentra un panel extensible con el nombre 'Filtrar'. Aparecerá un pequeño formulario con los posibles parámetros de búsqueda (ver figura A.1). Los datos se mostrarán en pantalla cuando se pulse el botón con el icono de la lupa.

Si es el administrador quien realiza esta acción, el formulario de búsqueda tendrá un campo añadido, profesor. Este cambio se puede apreciar en la figura A.2.

Figura A.1: *Filtros de búsqueda para un profesor*

Figura A.2: *Filtros de búsqueda para el administrador*

A.2. Exportar histórico

El botón 'Exportar históricos' (ver en la figura A.3) exportará en formato '.xlsx' el listado de históricos que haya resultado de una búsqueda. Tras unos segundos, se descargará en nuestro navegador con la información solicitada. No será posible exportar el fichero en caso de pulsar sin haber realizado ninguna búsqueda, la aplicación informará de ello y no exportará nada.

Histórico

Siglas/Asignatura	Año académico	Curso	Grupo	Cuatrimestre
[TP2] Tecnología de la programación 2	2019/2020	2º	E	2
[IN] Informática	2019/2020	1º	M1	2
[MAR] Métodos algorítmicos en resolución de problemas	2019/2020	3º	V	1
[TAIS] Técnicas algorítmicas en ingeniería del software	2019/2020	3º	E	1
[PD] Programación Declarativa	2019/2020	4º		1
[MAR2] Métodos algorítmicos en resolución de problemas 2	2019/2020	3º		2
[DA] Diseño de algoritmos	2019/2020	3º	D	1

Filtrar

Exportar historicos

Mostrar

☐ Profesor
☐ Código
☒ Nombre
☒ Curso
☒ Año académico
☒ Grupo
☒ Cuatrimestre
☐ Titulación
☐ Departamento

Figura A.3: Botón para exportar históricos

A.3. Importar histórico

Esta funcionalidad solo puede realizarla el administrador. Una vez se pulse el botón 'Importar histórico' en la sección 'Histórico', la aplicación mostrará otra vista (ver figura A.4). En ella habrá un modelo de histórico para poder importar correctamente los datos. El funcionamiento es trivial: pulsa sobre el botón 'Seleccione un archivo', selecciona el histórico del curso académico que quiera importar y la importación se realizará.

Es importante recordar que los archivos que se importen deben almacenar únicamente los datos de un año académico en concreto. En caso de que en el histórico la aplicación haya guardados registros de ese año, la función los sobrescribirá.

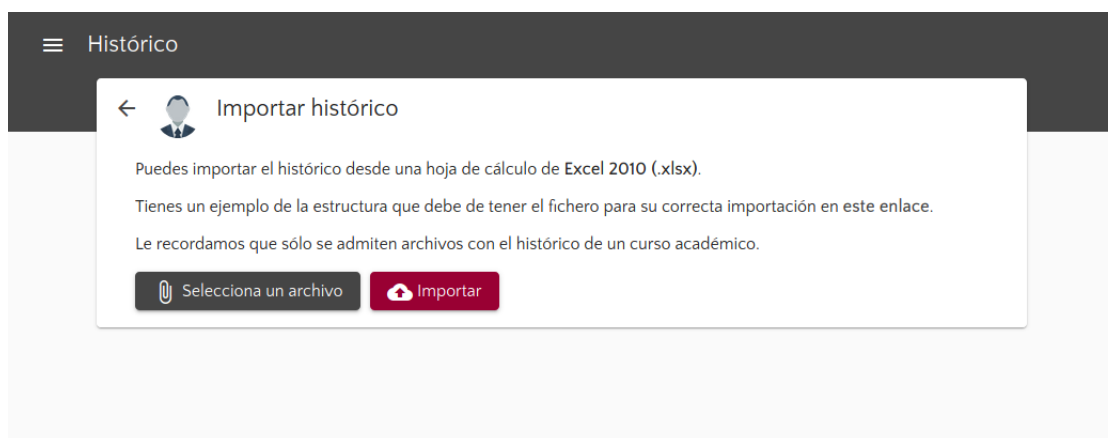


Figura A.4: *Pantalla para importar históricos*

A.4. Añadir reserva

Para realizar esta acción hay que navegar la vista 'Reserva/Interés'. Una vez aquí, se mostrará una lista con las asignaturas reservables (ver figura A.5). Para solicitar la reserva de una asignatura, bastará con pulsar sobre el botón 'Añadir' que habrá en la fila de la asignatura elegida.

Reserva docencia			
Reservar	Siglas/Asignatura	Curso	Cuatrimestre
Añadir	[MAR] Métodos algorítmicos en resolución de proble	3º	1
Añadir	[TAIS] Técnicas algorítmicas en ingeniería del softwar	3º	1
Añadir	[PD] Programación Declarativa	4º	1
Añadir	[MAR2] Métodos algorítmicos en resolución de proble	3º	2
Añadir	[DA] Diseño de algoritmos	3º	1
Resto de asignaturas			
Añadir	[ABBD] Administración de bases de datos	2º	1
Añadir	[ABD] Ampliación de bases de datos	3º	2
Añadir	[ABD] Ampliación de bases de datos	3º	2
Añadir	[ASCD] Análisis de sistemas concurrentes y distribui		2

+ Reservas
+ Muestras de interés
Filtrar
Mis reservas
Mostrar
☒ Nombre
☒ Curso
☒ Cuatrimestre
☐ Departamento

Figura A.5: *Lista de asignaturas reservables*

Al pulsar, aparecerá un *pop-up* en la pantalla con el objetivo de confirmar la decisión. En él se podrá añadir un comentario que posteriormente el administrador leerá. Se puede observar el *pop-up* en la figura A.6.

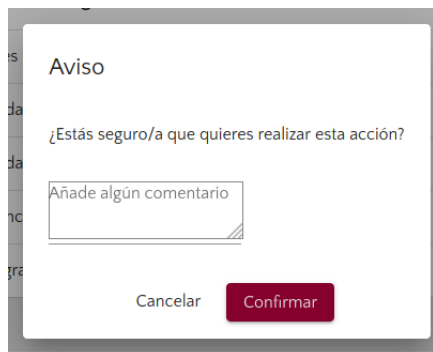


Figura A.6: *Pop-up de confirmación*

Para comprobar que efectivamente se ha añadido la reserva al sistema, a parte de aparecer la fila de la asignatura reservada de otro color, en el panel 'Mis reservas' se mostrarán las reservas realizadas por el profesor (ver figura A.7).

Reserva docencia

Reservar	Siglas/Asignatura	Curso	Cuatrimestre
Resto de asignaturas			
Añadir	[OIM] Organización de la Información y Metadatos	1º	1
Eliminar	[PGP] Planificación y Gestión de Proyectos	1º	2
Eliminar	[PLG] Procesadores de Lenguajes	4º	2
Añadir	[PCOM] Programación Competitiva	Opt.	1
Añadir	[PC] Programación Concurrente	3º	2
Añadir	[PC] Programación Concurrente	4º	2
Añadir	[PD] Programación Declarativa	3º	1
Añadir	[PD] Programación Declarativa	4º	1
Añadir	[PRDE] Programación Declarativa		

+ Reservas

+ Muestras de interés

Filtar

Mis reservas

- SGE2. Cuatri.: 2.
- PGP. Cuatri.: 2.
- PLG. Cuatri.: 2.

Mostrar

☒ Nombre
 ☒ Curso
 ☒ Cuatrimestre
 ☐ Departamento

Se ha guardado la reserva correctamente Cerrar

Figura A.7: *Lista de asignaturas reservables tras añadir alguna reserva*

A.5. Eliminar reserva

Para eliminar una reserva simplemente se tiene que pulsar en el botón 'Eliminar' de la asignatura sobre la que recae la reserva, que se verá de un color más oscuro que los demás. Una vez pulsado el botón, se habrá eliminado. En la figura A.8 podemos observar el resultado de eliminar una reserva.

The screenshot shows a web interface titled 'Reserva docencia'. It features a table of reservable subjects with columns: 'Reservar', 'Siglas/Asignatura', 'Curso', and 'Cuatrimestre'. The row for '[PGP] Planificación y Gestión de Proyectos' is highlighted, and the 'Eliminar' button is visible. A confirmation message 'Se ha eliminado la reserva correctamente' is shown at the bottom of the table. To the right, there are panels for '+ Reservas', '+ Muestras de interés', 'Filtar', 'Mis reservas' (listing 'SGE2. Cuatri.: 2.' and 'PGP. Cuatri.: 2.'), and 'Mostrar' (with checkboxes for 'Nombre', 'Curso', 'Cuatrimestre', and 'Departamento').

Reservar	Siglas/Asignatura	Curso	Cuatrimestre
Añadir	[OIM] Organización de la Información y Metadatos	1º	1
Eliminar	[PGP] Planificación y Gestión de Proyectos	1º	2
Añadir	[PLG] Procesadores de Lenguajes	4º	2
Añadir	[PCOM] Programación Competitiva	Opt.	1
Añadir	[PC] Programación Concurrente	3º	2
Añadir	[PC] Programación Concurrente	4º	2
Añadir	[PD] Programación Declarativa	3º	1
Añadir	[PD] Programación Declarativa	4º	1
Añadir	[PRDE] Programación Declarativa		
Añadir	[PD] Programación Declarativa		

Figura A.8: Lista de asignaturas reservables tras eliminar una reserva

A.6. Ver reservas

En la propia pantalla donde se añaden y se eliminan las reservas, a la derecha, se encuentran una serie de paneles y botones. En el panel de nombre 'Mis reservas', se muestran todas las reservas realizadas por el profesor. De estas reservas, se detallan: las siglas y el cuatrimestre de la asignatura reservada, el comentario de la reserva y el estado de ésta. Los 3 estados y por consiguiente, los 3 iconos que pueden aparecer son: aceptada (icono de check), rechazada (icono de cruz) e indeciso (icono de interrogante). En la figura A.9 se pueden observar reservas en los 3 estados.

Reserva docencia

Reservar	Siglas/Asignatura	Curso	Cuatrimestre
Añadir	[EDA] Estructuras de datos y algoritmos	2º	1
Añadir	[TP1] Tecnología de la programación 1	2º	1
Añadir	[GIW] Gestión de la Información en la Web	Opt.	1
Eliminar	[FAL] Fundamentos de algoritmos	2º	1

Resto de asignaturas

Añadir	[ABBD] Administración de bases de datos	2º	1
Añadir	[ABD] Ampliación de bases de datos	3º	2
Añadir	[ABD] Ampliación de bases de datos	3º	2
Añadir	[ASCD] Análisis de sistemas concurrentes y distribui		2
Añadir	[AERR] Análisis estático de programas y resolución c		1

+ Reservas

+ Muestras de interés

Filtar

Mis reservas

- ED. Cuatri.: 2.
- ED. Cuatri.: 2.
- FSW. Cuatri.: 1.
- IS2. Cuatri.: 2.
- FAL. Cuatri.: 1.
- GBD. Cuatri.: 1.
- PCOM. Cuatri.: 1.

Mostrar

☒ Nombre
 ☒ Curso

Figura A.9: Panel con las reservas realizadas por un profesor

A.7. Administrador - Aceptar reserva

Esta funcionalidad solo puede realizarla el administrador. Para poder aceptar una reserva debemos pulsar el botón 'Confirmar' que aparece en la vista principal de las reservas.

Este botón nos lleva a la pantalla donde el administrador gestiona las reservas realizadas por el profesor (ver figura A.10). Para ello, dispone de un *combobox* con todos los profesores para poder ver cada una de las reservas realizadas por ellos. Para cada reserva aparecen 2 botones, el primero de ellos sirve para aceptar la reserva y el segundo para rechazarla.

Reserva docencia

← Profesor: Carlos Gregorio...

Siglas	Cuatri	Comentario	Confirmar
SCE2	2		✓ ✕
PGP	2		✓ ✕
MAR2	2		✓ ✕
GIW	1		✓ ✕

+ Reservas

+ Muestras de interés

Alertas (8)

Figura A.10: *Lista de reservas de un profesor*

Aceptar una reserva es un punto de inflexión en el proceso, ya que dependiendo del contexto en el que se encuentre, los eventos que sigues son distintos. Lo que marca ese desencadenamiento de eventos es el número de grupos disponibles que tenga la asignatura relacionada. El número de grupos disponibles de una asignatura es la diferencia entre el número de grupos totales y el número de reservas para esa asignatura existan en el sistema.

- Si el número de grupos disponibles es mayor que 0, la reserva será aceptada. Será responsabilidad del profesor elegir el grupo en su turno de elección de docencia.
- Si el número de grupos disponibles es menor o igual a 0, la reserva será aceptada. En este caso, al estar todos los grupos solicitados por reservas, es el administrador quien tiene la responsabilidad de asociar el grupo de la asignatura asociada a la reserva. Para ello, la aplicación responde mostrando el formulario que se ve en la figura A.11.

Aviso

¿Estás seguro/a que quieres realizar esta acción?

El número de reservas de esta asignatura es igual o superior al número de grupos que ésta tiene, por lo que tendrá que elegir el grupo el administrador. Si no ha hablado con el profesor que ha realizado la reserva, hágalo antes de continuar.

Elige grupo y horario

Grupo: - Horario: M: 14:00:00-15:50:00/J: 14:00:00-15:50:00/ ▼ Creditos seleccionados 10

Añade algún comentario

Figura A.11: *Formulario para asignar una asignatura a la reserva*

En caso de que se trate de una asignatura divisible, una vez se elija el grupo, aparecerá un input donde se pedirá el número de créditos reservados. Además, el administrador podrá agregar un comentario a la reserva.

- Si todos los grupos de la asignatura están elegidos, la reserva seleccionada pasará directamente a estado rechazada.

En la figura A.12 se puede observar el resultado que se muestra por pantalla al realizar esta acción. La reserva aceptada aparecerá con un sombreado verde, indicando la confirmación de la misma.

Reserva docencia

← Profesor: Carlos Gregorio...

Siglas	Cuatri	Comentario	Confirmar
SCE2	2		↻
PCP	2		✓ ✕
MAR2	2		✓ ✕
GIW	1		✓ ✕

+ Reservas

+ Muestras de interés

Alertas (7) ▾

Se ha guardado la reserva correctamente

Cerrar

Figura A.12: *Lista de reservas tras aceptar una reserva*

A.8. Administrador - Rechazar reserva

Como se ha explicado en la sección anterior, existen dos botones para una reserva que su estado sigue siendo indeciso. Si la acción que se quiere realizar es rechazar la reserva, hay que pulsar el botón que tiene el icono de la cruz. Acto seguido aparecerá un *pop-up* con una caja de comentarios para confirmar la acción. En la caja, el administrador podrá añadir cualquier mensaje que quiera hacer llegar al profesor propietario de la reserva. En la figura A.13 se observa la consecuencia de rechazar una reserva. La reserva aparecerá con un sombreado gris, señal de que ha sido rechazada.

Reserva docencia

← Profesor: Carlos Gregorio...

Siglas	Cuatri	Comentario	Confirmar
SCE2	2		↺
PGP	2		✓ ✕
MAR2	2		✓ ✕
GIW	1		↺

+ Reservas
+ Muestras de interés
Alertas (7)

Se ha guardado la reserva correctamente Cerrar

Figura A.13: *Lista de reservas tras rechazar una reserva*

A.9. Administrador - Deshacer reserva

Una vez se ha decidido sobre una reserva, aceptar o rechazar, en la columna 'Confirmar' aparece un botón con un icono de volver para atrás. Este botón alberga la funcionalidad de deshacer, es decir, devolver al estado de indeciso a la reserva, independientemente del estado del que viene. En las figuras A.14 y A.15 se pone de manifiesto el cambio que produce la acción deshacer una reserva. Observe que la asignatura PGP primero queda sombreada con fondo verde (A.14) y posteriormente con fondo blanco (A.15)

Reserva docencia

← Profesor: Carlos Gregorio...

Siglas	Cuatri	Comentario	Confirmar
SCE2	2		
PGP	2		
MAR2	2		✓
GIW	1		

+ Reservas

+ Muestras de interés

Alertas (5)

Se ha guardado la reserva correctamente [Cerrar](#)

Figura A.14: *Lista de reservas de un profesor antes de deshacer una reserva*

Reserva docencia

← Profesor: Carlos Gregorio...

Siglas	Cuatri	Comentario	Confirmar
SCE2	2		
PGP	2		✓
MAR2	2		✓
GIW	1		

+ Reservas

+ Muestras de interés

Alertas (6)

Se ha guardado la reserva correctamente [Cerrar](#)

Figura A.15: *Lista de reservas de un profesor después de deshacer una reserva*

A.10. Añadir muestra de interés

Para realizar esta acción hay que navegar a la misma vista que para añadir una reserva. Una vez aquí, en la parte superior derecha, encontramos dos botones: 'Reservas' y 'Muestras

de interés'. Estos dos botones son los encargados de cambiar los datos que se muestran en la tabla.

Pulsamos sobre 'Muestras de interés'. Se mostrará la lista de todas las asignaturas de interés (ver figura A.16). Para mostrar interés por una asignatura, se pulsa sobre el botón 'Añadir' que habrá en la fila de la asignatura elegida.



Interés	Siglas/Asignatura	Curso	Cuatrimestre
Añadir	[MDL1] Matemática Discreta y Lógica Matemática 1	1º	1
Añadir	[MDL2] Matemática Discreta y Lógica Matemática 2	1º	2
Añadir	[TP1] Tecnología de la programación 1	2º	1
Añadir	[TP2] Tecnología de la programación 2	2º	2
Añadir	[BD] Bases de datos	2º	1
Añadir	[NSQ] Bases de datos NoSQL	Opt.	2
Añadir	[MIN] Minería de datos y el paradigma Big Data	Opt.	2
Añadir	[GIW] Gestión de la Información en la Web	Opt.	1
Añadir	[PR] Programación con restricciones	Opt.	2
Añadir	[TSW] Testing de software	Opt.	1

+ Reservas

+ Muestras de interés

Filtrar

Mis intereses

Mostrar

- ☒ Nombre
- ☒ Curso
- ☐ Grupo
- ☒ Cuatrimestre
- ☐ Departamento

Figura A.16: *Lista de asignaturas de interés*

Al pulsar, aparecerá un *pop-up* en la pantalla con el objetivo de confirmar la decisión. En él se podrá añadir un comentario que posteriormente el administrador leerá. Para comprobar que efectivamente se ha añadido la muestra de interés al sistema, a parte de aparecer la fila de la asignatura reservada con el fondo color grisáceo (e6e6e6), en el panel 'Mis intereses' se mostrarán las asignaturas por las que ha mostrado interés el profesor (ver figura A.17).

Muestra de interés

Interés	Siglas/Asignatura	Curso	Cuatrimestre
Eliminar	[DABI] Diseño de algoritmos bioinspirados		1
Añadir	[BD] Bases de datos	Opt.	1
Eliminar	[BIGD] Datos masivos: Big Data	Opt.	2
Añadir	[BBDD] Bases de datos		2
Añadir	[HIBD] Herramientas Informáticas para Big Data		2
Añadir	[SWES] Software estadístico		1
Añadir	[DWEB] Diseño web, tecnologías de Internet y redes		1
Añadir	[CUBI] Computación ubicua y nuevos modelos de a		2
Eliminar	[NTSA] Nuevas tecnologías en la sociedad actual		1
Añadir	[FSW] Fundamentos de software		

+ Reservas

+ Muestras de interés

Filtrar

Mis intereses

- NTSA. Cuatri.: 1.
- DABI. Cuatri.: 1.
- BIGD. Cuatri.: 2.

Mostrar

☒ Nombre
 ☒ Curso
 ☐ Grupo
 ☒ Cuatrimestre
 ☐ Departamento

Se ha creado la muestra correctamente

Cerrar

Figura A.17: *Lista de asignaturas de interés tras mostrar interés por alguna*

A.11. Eliminar muestra de interés

Al igual que para eliminar una reserva, se tiene que pulsar en el botón 'Eliminar' de la asignatura sobre la que recae el interés, que se verá de un color más oscuro que los demás. Una vez pulsado el botón, se habrá eliminado. El resultado se puede observar en la figura A.18.

Muestra de interés

Interés	Siglas/Asignatura	Curso	Cuatrimestre
Eliminar	[DABI] Diseño de algoritmos bioinspirados		1
Añadir	[BD] Bases de datos	Opt.	1
Añadir	[BIGD] Datos masivos: Big Data	Opt.	2
Añadir	[BBDD] Bases de datos		2
Añadir	[HIBD] Herramientas Informáticas para Big Data		2
Añadir	[SWES] Software estadístico		1
Añadir	[DWEB] Diseño web, tecnologías de Internet y redes		1
Añadir	[CUBI] Computación ubicua y nuevos modelos de aj		2
Eliminar	[NTSA] Nuevas tecnologías en la sociedad actual		1
Añadir	[FSW] Fundamentos de softwa		

+ Reservas

+ Muestras de interés

Filtar

Mis intereses

- NTSA. Cuatri.: 1.
- DABI. Cuatri.: 1.

Mostrar

- ☒ Nombre
- ☒ Curso
- ☐ Grupo
- ☒ Cuatrimestre
- ☐ Departamento

Muestra de interés eliminada correctamente

Cerrar

Figura A.18: *Lista de asignaturas de interés tras eliminar una muestra de interés*

A.12. Ver muestras de interés

En la pantalla donde se añaden y se eliminan las muestras de interés, a la derecha, se encuentra el panel 'Mis intereses' donde se muestran todas las asignaturas por las que ha mostrado interés el profesor (ver figura A.19). Para cada una, se detallan: las siglas y el cuatrimestre de la asignatura de interés, el comentario y el estado del interés. Los iconos que se muestran son los mismos que para las reservas.

Muestra de interés

Interés	Siglas/Asignatura	Curso	Cuatrimestre
Eliminar	[DABI] Diseño de algoritmos bioinspirados		1
Añadir	[BD] Bases de datos	Opt.	1
Añadir	[BIGD] Datos masivos: Big Data	Opt.	2
Añadir	[BBDD] Bases de datos		2
Añadir	[HIBD] Herramientas Informáticas para Big Data		2
Añadir	[SWES] Software estadístico		1
Añadir	[DWEB] Diseño web, tecnologías de Internet y redes		1
Añadir	[CUBI] Computación ubicua y nuevos modelos de aj		2
Eliminar	[NTSA] Nuevas tecnologías en la sociedad actual		1
Añadir	[FSW] Fundamentos de software		1

+ Reservas

+ Muestras de interés

Filtrar

Mis intereses

- NTSA. Cuatri.: 1.
- DABI. Cuatri.: 1.

Mostrar

☒ Nombre
 ☒ Curso
 ☐ Grupo
 ☒ Cuatrimestre
 ☐ Departamento

Figura A.19: Panel donde se listan las muestras de interés

A.13. Administrador - Aceptar muestra de interés

Para realizar esta acción, nos debemos situar la pantalla del administrador de la gestión de las reservas. Pulsamos sobre el botón 'Muestra de interés' de arriba a la derecha. Aparecerán las asignaturas por las que el profesor seleccionado en el *combobox* ha mostrado interés. En la figura A.21 se observa la lista de muestras de un profesor.

Muestra de interés

← Profesor: Carlos Gregorio...

Siglas	Cuatri	Comentario	Confirmar
NTSA	1		✓ ✕
DABI	1		✓ ✕

+ Reservas

+ Muestras de interés

Alertas (6)

Figura A.20: Lista de muestras de interés de un profesor

Las asignaturas de interés suelen tener un solo grupo, dado que son de máster o docencia en lengua inglesa. Es por ello que para las muestras de interés siempre aparecerá el formulario de elección de grupo. El administrador será quien decida el grupo y el número de créditos, si se trata de una asignatura divisible. Este formulario no saldrá por pantalla cuando todos los grupos de esa asignatura estén asignados a algún profesor. Tras rellenar todos los campos, la tabla donde se exponen las muestras de interés se verá como en la figura A.21. Utilizando el mismo mecanismo que para las reservas, la muestra de interés automáticamente tendrá el estado de rechazada.

Siglas	Cuatri	Comentario	Confirmar
NTSA	1		✓
DABI	1		↻

Figura A.21: *Lista de muestras de interés tras aceptar una de ellas*

A.14. Administrador - Rechazar muestra de interés

Para rechazar la muestra de interés, al igual que pasa con las reservas, hay que pulsar el icono de la cruz. Esto provocará que salte un *pop-up* con una caja de comentarios para confirmar la acción. En la caja, el administrador podrá añadir cualquier mensaje que quiera hacer llegar al profesor que mostró interés por la asignatura. El resultado de realizar esta acción se puede ver en la figura A.22.

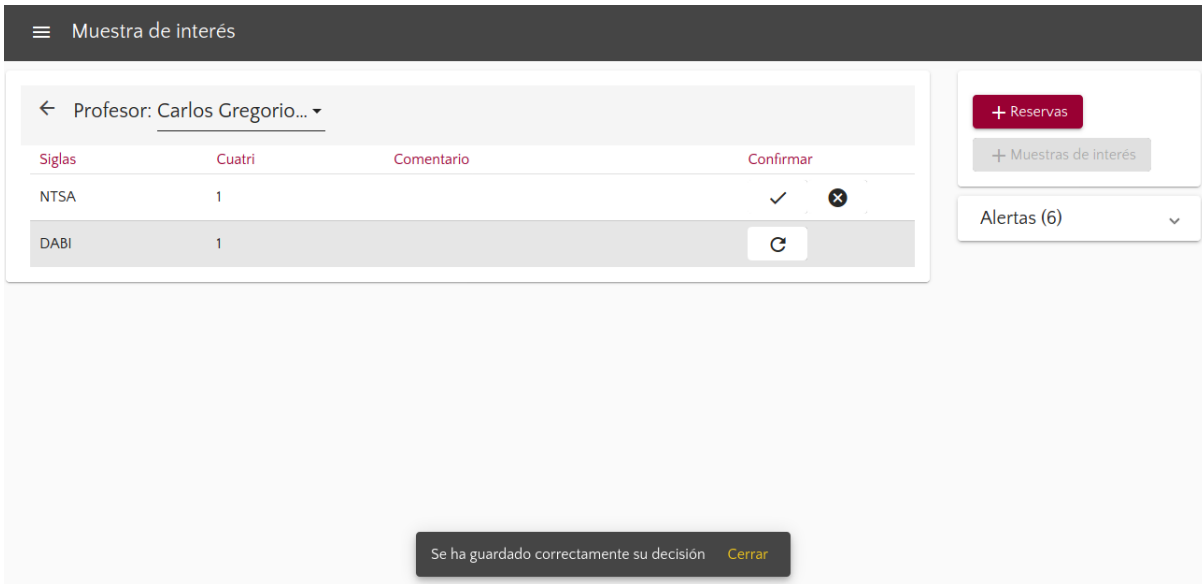


Figura A.22: *Lista de muestras de interés tras rechazar una de ellas*

A.15. Administrador - Deshacer muestra de interés

Se trata de la misma funcionalidad que la presentada en la sección A.9 de este apéndice. En las figuras A.23 y A.24 se representa el cambio que produce deshacer una muestra de interés.

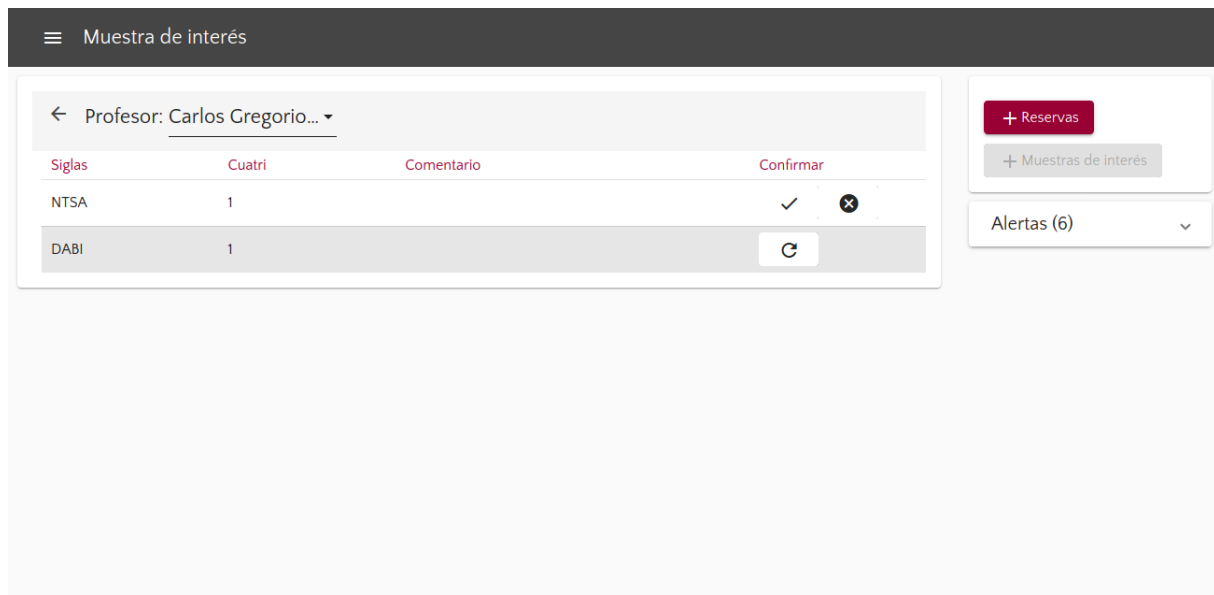


Figura A.23: *Lista de muestras de interés antes de deshacer una de ellas rechazar una reserva*

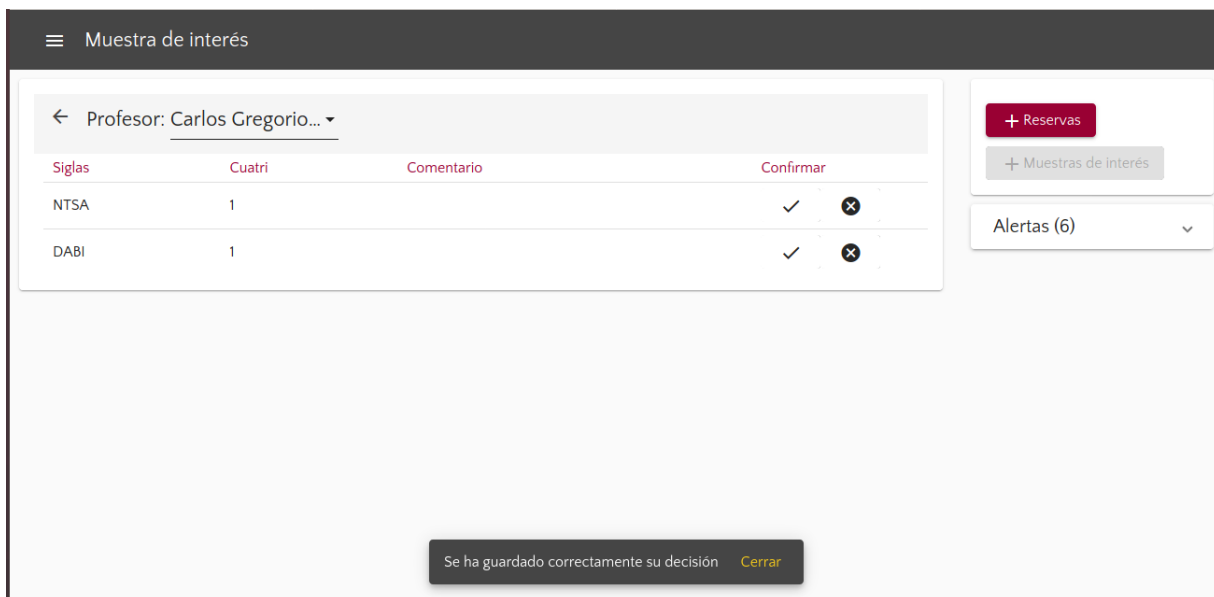


Figura A.24: *Lista de muestras de interés tras deshacer una muestra*

Apéndice B

Guía de instalación

B.1. Instalación de la aplicación web

Para la instalación de la aplicación, se utilizará un servidor Apache en un sistema Linux. Los requisitos necesarios para la correcta compilación del código son instalar Nodejs en su versión 8.x o 10.x y el gestor de paquetes npm. Para ello, introduciremos los siguientes comandos:

```
$ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.34.0/install.sh | bash  
$ command -v nvm
```

Si la salida del último comando es 'nvm', se habrá instalado correctamente el gestor de versiones. En este caso, mi versión de Nodejs es 0.7.0 y de npm, estoy utilizando la versión 6.11.3.

El siguiente paso es instalar todas las dependencias del proyecto Angular. Para ello, introducimos por línea de comandos:

```
$ cd tfg-angular  
$ npm install
```

Importante no utilizar sudo en ningún momento, ya que esto generaría problemas de permisos y no podríamos lanzar la aplicación correctamente.

B.2. Instalación de la API

Esta instalación se realizará en una máquina con el sistema operativo Ubuntu 18.04 LTS. Cualquier sistema operativo basado en la arquitectura Debian o similares podrá seguir esta instalación al detalle. Para su correcta instalación, es obligatorio cumplir una serie de requisitos: instalar el intérprete de Python 3, pip (administrador de paquetes), Apache2 y *virtualenv*.

El primer paso es actualizar el sistema:

```
$ sudo apt-get update -y
```

```
$ sudo apt-get upgrade -y
```

```
$ python3 --version
```

El último comando sirve para comprobar que el sistema cuenta con la tercera versión del intérprete de Python. A continuación, se instala *virtualenv*, la herramienta que permitirá crear entornos virtuales de Python.

```
$sudo pip3 install virtualenv
```

```
$virtualenv --python python3 DocenciaEnv
```

Se arranca el entorno virtual:

```
$source DocenciaEnv/bin/activate
```

Se procede a instalar todas las dependencias necesarias para el correcto funcionamiento de la API:

```
$pip install Django
```

```
$pip install djangorestframework
```

```
$pip install django-cors-headers
```

```
$pip install django-filter
```

```
$pip install django-sendgrid-v5
```

```
$pip install djangorestframework-simplejwt
```

```
$pip install drf-nested-routers
```

```
$pip install openpyxl
```

```
$pip install coreapi
```

```
$pip install coreschema
```

Los siguientes comandos crearán la estructura de la base de datos.

```
$ cd TFG_docencia
```

```
$ python manage.py makemigrations
```

```
$ python manage.py migrate
```

```
$ python manage.py loaddata departamentos_fixture.json
```

Es necesario agregar algunos permisos sobre algunos directorios para que todo funcione correctamente. Se incluye un *script* en el proyecto que facilita la tarea, aunque siempre se podrán introducir uno a uno por línea de comandos:

```
$ chmod g+w TFG_docencia
```

```
$ sudo chown :www-data TFG_docencia
```

```
$ chmod g+w TFG_docencia/db.sqlite3
```

```
$ sudo chown :www-data TFG_docencia/db.sqlite3
```

```
$ chmod g+w -R TFG_docencia/uploads
```

```
$ sudo chown -R :www-data TFG_docencia/uploads
```

```
$ chmod g+w -R TFG_docencia/asignaturas/fixtures
```

```
$ sudo chown -R :www-data TFG_docencia/asignaturas/fixtures
```

```
$ chmod g+w -R TFG_docencia/departamentos/fixtures
```

```
$ sudo chown -R :www-data TFG_docencia/departamentos/fixtures
```

```
$ chmod g+w -R TFG_docencia/profesores/fixtures
```

```
$ sudo chown -R :www-data TFG_docencia/profesores/fixtures
```

Además, se debe crear un superusuario para que pueda importar los profesores y las asignaturas. *\$ python manage.py createsuperuser*

Se necesita volcar toda la información estática de la aplicación. Para ello,

```
$ python manage.py collectstatic
```

Finalmente, se configura el servidor web. Introducimos las siguientes cadenas por línea

de comandos:

```
$ sudo apt-get install apache2-dev -y
```

```
$ sudo apt-get install apache2-mpm-worker -y
```

```
$ sudo apt-get install libapache2-mod-wsgi-py3
```

Después de instalar el servidor web, hay que editar el archivo de configuración para servir la API en Internet. El archivo se muestra en la figura B.2.

```
$ nano /etc/apache2/sites-available/000-default.conf
```

```
<VirtualHost *:80>
    ServerName docenciafdi.me
    ServerAlias www.docenciafdi.me
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    Alias /static /home/TFG_docencia/static
    <Directory /home/user/TFG_docencia/static>
        Require all granted
    </Directory>
    <Directory /home/user/TFG_docencia/TFG_docencia>
        <Files wsgi.py>
            Require all granted
        </Files>
    </Directory>
    WSGIDaemonProcess TFG_docencia python-
    path=/home/user/TFG_docencia python-
    home=/home/user/DocenciaEnv
    WSGIProcessGroup TFG_docencia
    WSGIScriptAlias /
    /home/user/TFG_docencia/TFG_docencia/wsgi.py
    WSGIPassAuthorization On
</VirtualHost>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Figura B.1: Archivo de configuración de la API

Para acabar, reiniciamos el servidor para que se integren todos los cambios realizados:

```
$ sudo service apache2 restart
```